SYSTEM IDENTIFICATION FOR DEVELOPMENT OF DISTRIBUTED ELECTRIC PROPULSION VERTICAL TAKE-OFF VEHICLES

BY

ANDREW PATTERSON

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Professor Naira Hovakimyan

Abstract

This thesis presents a complete procedure for estimating the inertia matrix of a multi-rotor vertical take-off and landing vehicle. The procedure starts with an overview of vehicle dynamics. An estimator and a controller are introduced based on these dynamics, which can be tuned to the desired performance specifications. Once the vehicle is flying, data is collected and system identification starts. Several methods of inertia estimation are covered that are simple to implement but still take into account the constraints of symmetry in the inertia matrix. A new method of calculating the inertia is then introduced, which takes in the additional constraint of non-negativity. Each of these methods is compared using multi-sine test inputs generated to optimize performance without driving the system unstable. Finally, both simulation and implementation results are provided.

Though inertia estimation is common, the method presented in this thesis is cost effective, requiring almost no extra measurement equipment; it is algorithmically simple, using only the necessary computations; and this thesis completely describes the steps necessary to implement and estimate a vehicle's inertia.

Acknowledgements

I would first like to thank my advisor, Professor Naira Hovakimyan, for her dedication to perfection and her commitment to all of her students.

I also thank all my colleagues in the ACRL for making the late nights in the lab feel less like work: Thiago by making sure no one has to eat alone, Arun with his constant encouragement and Mitchell who is always ready for anything.

I am forever grateful for my family, who I would not be here without, and for my fiancée, Kate, who always encourages me to dream of everywhere I can go.

Table of Contents

Chapter 1

Introduction

1.1 Background

The transportation industry faces the challenge of increasing the speed and efficiency of transport without overusing the existing infrastructure [1]. In the past century, aircraft have revolutionized the transportation industry. The benefits have been seen on the industrial scale, with air transportation providing another option for the rapid transport of people and goods. Aircraft have seen widespread use since their invention as they only require infrastructure at the endpoints and take advantage of the volume of airspace instead of being constrained to the surface of the Earth. However, scaling these benefits down to the individual level has long been a goal out of reach. The first major challenge is endpoint infrastructure, which is often extensive, requiring a runway and fuel storage. This infrastructure demands space and is costly, which prevents it from being used on an individual level. In rural areas, this challenge is overcome with airparks where a subdivision shares an airfield.

With advancements in battery technology, electric propulsion and aircraft control, distributed electric propulsion aircraft (DEPA) are being designed to fill the niche role that traditional aircraft have been unable to fill, particularly executing short-range flights with low infrastructure and setup costs. Distributed electric propulsion relies on brushless motors

for propulsion, which scale in both efficiency and manufacturability. One of the most well known versions of this aircraft is the multirotor. Multirotors have widespread application, ranging in size from toys smaller than the size of a hand to recently developed vehicles that can carry a person [2].

The multirotor is one of the simpler cases of DEPA, but more complex versions are coming out daily that extend the use cases, trying to take full advantage of the simplicity of electric motors and power transmission. Of particular interest are tilt-wing and tilt-rotor vehicles, which no longer have to move entire turbines or drive trains but can shift the angle of a wing with a slip-ring or slack in the power transmission lines. These types of vehicles offer the maneuverability of multi-rotors with the long-distance efficiency of a fixed wing aircraft. Companies such as Airbus and Uber have announced interest and produced concepts of these craft in [3] and [4].

Since each of these designs is unique, each requires new research to characterize the system for control. This characterization requires extensive testing of both individual components and the system as a whole. Taking advantage of the scalability of the actuators, flight-testing can be often performed at different scales to mitigate the cost of failure without loss of accuracy. The aerodynamics can be scaled using techniques found in [5] and the electrical propulsion can be scaled based on power requirements. This motor scaling is cheaper than jet engine scaling. These benefits make small scale testing cost effective and useful for control design. One example of this is the NASA GL-10, which is a tilt-wing VTOL aircraft and has both a 50% scale testing platform with a 10-foot wingspan and 35% scale testing platform with a 7-foot wingspan for different technology tests. [6].

1.2 Challenges

Working with scaled models can allow design choices to be validated and mistakes to be less expensive. However, these scaled models require the same amount of system identification as the full system. This testing could include costly thrust testing, inertia measurement, and aerodynamic characterization using a wind tunnel. In many cases, these tests are only performed on the final design, not on scaled prototypes. If the prototypes are small, benchtop testing is a viable option, but for larger craft, more infrastructure may be necessary. For example, even the 35% scale model of the GL-10 would require a room-sized test stand to calculate the inertia.

1.3 Experimental Overview

The goal of this experiment is to show that a hand-tuned estimator and controller can be used to estimate the parameters necessary to design a model-based controller for a VTOL aircraft. The demonstration platform is a quadrotor, which is controlled externally with attitude angle targets provided by a human operator. The estimator and controllers are tuned to meet tracking performance requirements using only onboard sensor measurements. The control input is augmented with test inputs to generate rich information for system identification. In the data analysis step, two methods of system identification are presented in this thesis. The first is the traditional least squares method found in [7]. The second system identification procedure uses the non-negative least squares method, found in [8]. A flowchart outlining the experimental steps is shown in Figure 1.1.

1.4 Related Work

An overview of system identification techniques can be found in [9]. A variety of different methods of quadrotor system identification have been described in recently published papers.

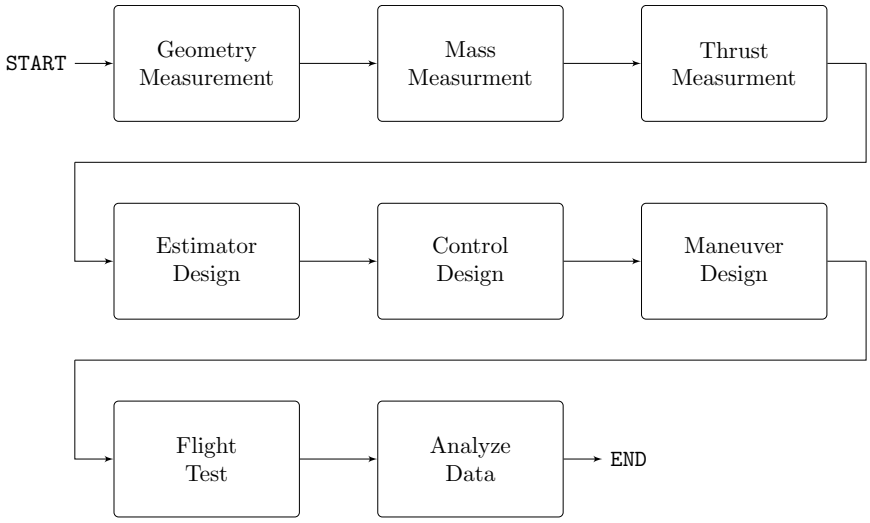

Figure 1.1: System identification process flow chart.

For example, [10] and [11] use external measurement equipment to identify parameters, and [12] uses grey-box system identification but does not describe the process in detail. Inertia identification in satellites is covered theoretically in [7] using different filtering techniques to estimate angular acceleration. We will not focus on the identification of the fixed-wing aerodynamic coefficients since these are well described in [13] and [14].

1.5 Thesis Overview

Since hand flight of a vehicle with a tuned controller is often the first step in vehicle development, the purpose of this thesis is to outline a procedure for control system development with limited knowledge of system dynamics so that the model can be identified and refined. The design and modeling process is broken down into the chapters of this thesis.

In chapter 2, the equations of motion for a rigid body are described and simplified, and the known and unknown parameters are listed. The test platform is introduced, and the mathematical background necessary for the controller and observer design is covered.

In chapter 3, the state estimator is introduced and demonstrated. The estimator is responsible for approximating the attitude of the rigid body. The estimator described acts directly on the vehicle's configuration space, allowing it to be free of uncertainties due to linearization and free of singularities. In addition to an overview of the observer dynamics, some rules for parameter selection and tuning are provided. Performance results are shown in simulation.

In chapter 4, the control objective is described in detail and two controllers will be designed and tuned for the vehicle. The first is the standard cascaded PID structure, which is included since this is often the base controller for commercial vehicles. The second, and preferred, controller is constructed and tuned in the second part of the chapter. It is a non-cascaded controller and operates directly on the system configuration space, which allows it to work in any orientation. As with the estimator, a guideline for gain selection is suggested and the results are shown in simulation.

Chapter 5 introduces the steps the for system identification procedure. These steps include thrust curve fitting, a description of least squares methods, and test input generation.

Chapter 6 assumes we have a flyable aircraft for which we generate test inputs and flight patterns to identify the unknown parameters of the system. The challenges of identification and possible remedies are discussed. Data from both simulations and flight tests are shown.

Chapter 2

Vehicle Dynamics

2.1 Overview of System

Although the demonstration platform is a quadrotor, the equations described below will work for any propeller-based DEPA with multiple rotors. We will investigate the kinematic and dynamic models and then show where simplifications occur. A full description of the quadrotor dynamics and aerodynamic effects can be seen in [15], [16] and [17]. The quadrotor has four-thrust producing propellers spaced uniformly around the body of the vehicle. The thrust they generate allows the vehicle to fly. The platform rolls and pitches based on the difference between the thrust of the motors. The vehicle changes yaw angle based on the difference in the motor speeds of counter-rotating propellers that are creating a torque differential. A mathematical description of this motion is described in a later section.

For any VTOL DEPA, the equations of motion will have a similar structure, since the equations assume only that the propellers are attached to a rigid body. The equations could be extended by considering the propellers' individual effects and changing the locations at which the forces are applied.

2.2 Assumptions

We make the following assumptions on the problem:

1. The quadrotor is rigid.
2. The quadrotor center of gravity is at the center of the body reference frame.
3. Nonlinear aerodynamic effects (e.g. blade flapping, induced drag, ground effect) are negligible.

2.3 Rotation Representation

To describe the state of a rigid body, position, linear velocity, orientation and angular velocity are needed. The position, linear velocity and angular velocity are represented in three-dimensional Euclidean space, $\mathbb{R}^3$. For the description of rotational motions, there are several options. In this thesis, a rotation is represented by the special orthogonal group, $\mathrm{SO}(n)$. This group can be represented as matrices that have orthogonal columns and rows; each column and row is a unit vector, and the determinant is positive one.

The rigid body has three rotational directions, so all rotations will be members of SO(3). The rotations are represented with a subset of 3×3 matrices. The estimator and one of the proposed controllers are directly designed on this space. As a result, both the controller and estimator are valid in any orientation.

2.3.1 Cross Product Operator

Define a skew-symmetric matrix operator $a_\times$, often called the hat operator, for any vector $a \in \mathbb{R}^3$ as

$$a_\times = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Then for any $q \in \mathbb{R}^3$ one has $a_\times q = a \times q$, which is a matrix representation for the vector cross product. This operation is particularly useful as our measurements of angular velocity are vectors in $\mathbb{R}^3$, but the rotation is represented as a matrix.

The inverse of this map is called the "Vee" operator. It is a mapping from SO(3) to $\mathbb{R}^3$. So for any $B \in \mathrm{SO}(3)$ this map is constructed as

$$B_\vee = \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix}_\vee = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

2.3.2 Time Derivative of a Rotation Matrix

Since the dynamic and kinematic equations use rotation matrices, it can be useful to define an infinitesimal change in a rotation matrix due to an arbitrarily small rotation. Let $R \in \mathrm{SO}(3)$ describe a rotation with respect to an arbitrary but fixed reference frame. Let an infinitesimal rotation about the x, y and z axis be given by

$$\mathrm{d}\phi \coloneqq \begin{bmatrix} \mathrm{d}\phi_x \\ \mathrm{d}\phi_y \\ \mathrm{d}\phi_z \end{bmatrix}. \tag{2.1}$$

Then the rotation matrix due to an infinitesimal rotation can be written as $R_{\mathrm{d}\phi} = \mathbb{I} + \mathrm{d}\phi_\times$, which does not depend on the order of rotation. The infinitesimal rotation matrix, $\mathrm{d}R$, can be written as the difference between the rotation due to the infinitesimal rotation, $R_{\mathrm{d}\phi}$, in the R frame and the original frame. This gives us

$$\begin{aligned}\mathrm{d}R &= R\,R_{\mathrm{d}\phi} - R \\ &= R(R_{\mathrm{d}\phi} - \mathbb{I}) \\ &= R\,\mathrm{d}\phi_\times.\end{aligned}$$

Angular velocity is defined as:

$$\Omega = \frac{\mathrm{d}\phi}{\mathrm{dt}} = \begin{bmatrix} \frac{\mathrm{d}\phi_x}{\mathrm{dt}} \\ \frac{\mathrm{d}\phi_y}{\mathrm{dt}} \\ \frac{\mathrm{d}\phi_z}{\mathrm{dt}} \end{bmatrix}, \tag{2.2}$$

and hence

$$\frac{\mathrm{d}R}{\mathrm{dt}} = \frac{1}{\mathrm{dt}} R\,\mathrm{d}\phi_\times = R\,\Omega_\times.$$

2.4 Coordinate System

Two reference frames for rigid body dynamics on a local scale are of primary interest. These will be right-handed coordinate systems with the standard x-y-z rotation order. The first coordinate frame $\mathcal{B}$ is attached to the centroid of the vehicle and aligned with a reasonable axis (usually the axis of symmetry). The z-axis is perpendicular to the top of the vehicle, pointing in the direction of propeller thrust.

The second coordinate frame $\mathcal{I}$ represents the inertial frame. It is attached locally to some point and aligned with z opposite the direction of the gravitational force and x-y aligned with either cardinal directions or some local feature such as x pointing towards the front of a flight area. In our case, the test vehicle will be flown in 'X' configuration, with the body

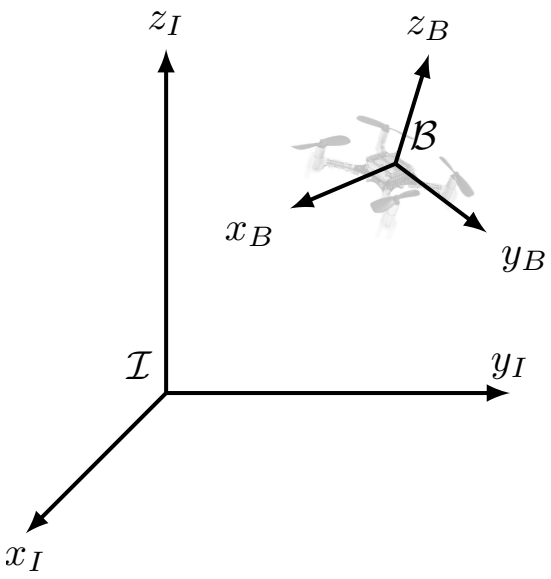

Figure 2.1: Inertial and Body Coordinate Frames

coordinates aligned with the x_B axis as shown in Figure 2.2. Other configurations, such as the cross configuration, use one set of propellers exclusively for roll and one for pitch. Additionally, the cited transport vehicles use an extension of this orientation, adding additional propellers parallel to the y_B as shown in Figure 2.3. These configurations allow movement that is more efficient after the transition to fixed-wing flight with the propellers perpendicular to the heading angle. To express the orientation measured in one coordinate frame in terms of another frame, a transformation is constructed. This transformation will be useful for expressing how the vehicle is moving in the inertial frame based on the measurements from sensors attached in the body.

To map from the inertial to the body frame, a series of rotations are created. This thesis uses the $Z-Y-X$ convention. This rotation order corresponds to a yaw-pitch-roll (ψ, θ, ϕ) rotation order. The rotation from the body to the inertial frame is denoted R_B^I. The

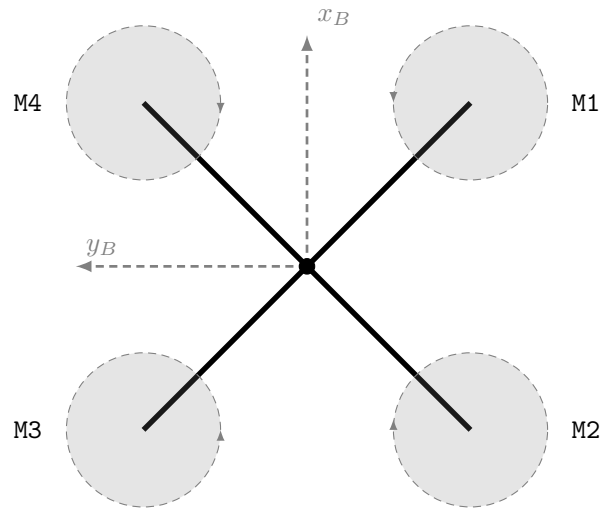

Figure 2.2: Coordinate and Body Frames

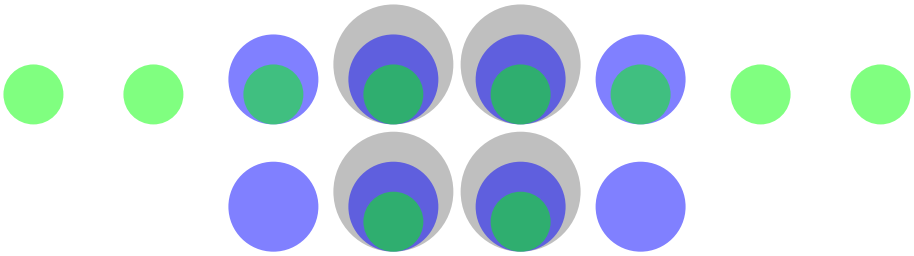

Figure 2.3: Vehicle Propeller distributions, quad in gray, Airbus Vahana in blue and GL-10 in green. Areas not to scale.

transformation is calculated from Euler angles such that

$$R_B^I = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - c_\phi s_\psi & s_\psi s_\phi + c_\psi c_\phi s_\theta \\ c_\theta s_\psi & c_\psi c_\phi + s_\psi s_\theta s_\phi & c_\phi s_\psi s_\theta - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}, \tag{2.3}$$

where $c_\alpha = \cos\alpha$ and $s_\alpha = \sin\alpha$. An example of the use of this rotation is transforming the body x-axis to world frame:

$$x_B = R_B^I [1,\, 0,\, 0]^\top. \tag{2.4}$$

2.5 Equations of Motion

The equations of motion are of particular importance to this thesis as they contain the parameters that describe the motion of the vehicle being identified. We will describe the dynamics generally as

$$\dot{\mathbf{x}} = g(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{2.5}$$

where $\mathbf{x}$ is the vehicle state and $\mathbf{u}$ is the control input. The state is given by $\mathbf{x} = \left[\xi^\top, \dot{\xi}^\top, \theta^\top, \omega^\top\right]^\top$. The vector $\xi = [x, y, z]^\top$ denotes the vehicle's position in the world frame, $\dot{\xi}$ represents the linear velocity, $\Theta = [\phi, \theta, \psi]^\top$ is the angular position in the world frame, and $\omega = [p, q, r]^\top$ is the angular rate in the body frame. The input $\mathbf{u}$ includes the force and moments that move the vehicle. The force, F, is aligned with the z_B axis, and the moments τ_x, τ_y, and τ_z are defined around the corresponding body axes. Using these moments we define the moment vector, $\boldsymbol{\tau} = [\tau_x, \tau_y, \tau_z]^\top$, and $\mathbf{u} = [F, \boldsymbol{\tau}^\top]^\top$.

2.5.1 Motor Force Mapping

Although we consider a force to be our system input, the command sent from the controller is an angular velocity command for the vehicle propeller, given by $\zeta_i \in [0, 1]$, where i is the index of the motor. We will model the relationship between the commanded throttle and the force f_i as a quadratic equation of the form:

$$f_{mi} = \begin{bmatrix} b_2 & b_1 & b_0 \end{bmatrix} \begin{bmatrix} \zeta_i^2 \\ \zeta_i \\ 1 \end{bmatrix}, \tag{2.6}$$

where b_0, b_1, b_2 are positive gain constants.

2.5.2 Motor Mapping

The mapping from the motor commands to the body force and torques, as mentioned before, is assumed to be a linear mapping given by

$$\begin{bmatrix} u_f \\ u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} f_{m1} \\ f_{m2} \\ f_{m3} \\ f_{m4} \end{bmatrix}, \tag{2.7}$$

which shows how the motors contribute to the body force and moments. As expected, the motors all contribute equally to the thrust of the aircraft. The torque command around the x-axis has a different sign based on which side of the x-axis the motors are on. Note that motors `M1` and `M2` are on the right side, while motors `M3` and `M4` are on the left, meaning they contribute opposing forces. The same is true for the pitch direction. The yaw command is based on the direction of rotation of propellers, where diagonally opposed motors spin in the same direction.

These commands are then scaled based on physical coefficients that map the force and torque commands to the force and torque produced by the motor propeller pair. These coefficients are the thrust coefficient, k_F, and the torque coefficient, k_Q. We scale these generated torques and forces by the distance from the axis of rotation based on the arm length, L. The body force and moments can be represented as

$$\begin{bmatrix} F \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{L}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{L}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & k_Q \end{bmatrix} \begin{bmatrix} u_f \\ u_x \\ u_y \\ u_z \end{bmatrix}. \tag{2.8}$$

2.5.3 Angular Acceleration

The angular acceleration of a rigid body under an external force can be described as

$$\dot{\omega} = \mathcal{J}^{-1} \left[\tau - \omega \times (\mathcal{J}\omega) \right], \tag{2.9}$$

where $\mathcal{J} \in \mathcal{R}^{3x3}$ is the symmetric inertia matrix. We can write Equation (2.9) in terms of ω as

$$\tau = \mathcal{J}\dot{\omega} + \omega \times \mathcal{J}\omega. \tag{2.10}$$

Since the inertia tensor is of particular interest, we write that $\mathcal{J}$ in Equation (2.9) is given by

$$\mathcal{J} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix}.$$

Moreover, the symmetry of $\mathcal{J}$ leads to a simplified form:

$$\mathcal{J} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix}. \tag{2.11}$$

We can model the linear acceleration of our vehicle as

$$\ddot{x} = -gz_I + \frac{F}{m} z_B. \tag{2.12}$$

For this thesis, the quantity of most interest is the mass of the vehicle m.

2.5.4 Summary of Equations

We describe the system dynamics with the following equations. The static speed to thrust mapping is given by

$$f_{mi} = \begin{bmatrix} b_2 & b_1 & b_0 \end{bmatrix} \begin{bmatrix} p_i^2 \\ p_i \\ 1 \end{bmatrix}.$$

The rotational dynamics can be formulated as

$$\dot{\omega} = \mathcal{J}^{-1} \left[\tau - \omega \times \mathcal{J}\omega \right].$$

The rotational kinematic update law based on the measured angular rotation is governed by

$$\dot{R} = R\omega_x.$$

Finally, the linear dynamics are given by

$$\ddot{x} = -gz_I + \frac{F}{m} z_B.$$

Chapter 3

Estimation

State estimation is performed using the explicit nonlinear complementary filter described in [18]. The estimator only has two tunable parameters in the most basic form. This observer is also easily scalable to include many different sensor inputs including magnetometer and external position estimates from motion capture systems such as VICON.

This section outlines the basis of the state observer, describes its implementation and tuning process, and then demonstrates its performance in an application.

3.1 Objective

The goal of estimation is to produce an estimate of the vehicle rotation $\hat{R}_B^I \in \mathrm{SO}(3)$ based on measurements from onboard sensors and knowledge of system dynamics. The sensor package used in this thesis is the MPU-9250. It includes both a three-axis gyroscope and a three-axis accelerometer, which measure angular velocity, denoted ω_{meas}, and linear acceleration, represented by a_{meas}.

3.2 Challenges

The two primary concerns with this type of sensor are noise and bias. Bias is inherent to the sensor measurement and is integrated during estimation, leading to a drift in the estimated state. Noise is a combination of external interference and internal imperfections. Given both accelerometer and gyroscope measurements, we have

$$\omega_{meas} = \omega + b_\omega + \nu_\omega$$
$$a_{meas} = R_I^B(\ddot{x} - g) + b_a + \nu_a,$$

where b_a and b_ω are the accelerometer, and gyroscope biases, g, is the effect of gravity on the accelerometer, and ν_ω, ν_a are additive noise terms. In the MPU-9250, the gyroscope bias is often significant and leads to drift in the angular estimation of the vehicle. Though large, it is slowly time varying. The acceleration bias is small and is not integrated. The signal-to-noise in the accelerometer measurement is small at low frequencies, since gravity dominates the measurement in most maneuvers. In the angular velocity measurement, noise is a significant contributor.

Assumptions

1. Bias does not change during a mission.

2. Angular velocity noise is zero mean white noise.

3.3 Estimator

The estimator is called the nonlinear complementary filter because of its structural similarity to linear complementary filters. A linear complementary filter attenuates the effect of bias in gyroscope measurements by applying a high-pass filter to angular velocity. Similarly, a low-pass filter is used to remove noise from the acceleration signal. These two filtered mea-

surements are then combined to reconstruct the estimated attitude. Similarly, this nonlinear estimator integrates the angular acceleration of the vehicle and includes correction terms based on the accelerometer measurement.

The estimate for the vehicle attitude is constructed with the following equations:

$$\dot{\hat{R}}_B^I = \hat{R}_B^I(\omega_{meas} - \hat{b}_\omega)_\times - \alpha \tag{3.1}$$

$$\dot{\hat{b}} = k_b\alpha \tag{3.2}$$

$$\alpha = \frac{k_a}{g^2}(\hat{R}_I^B z_I \times a_{meas})_\times, \tag{3.3}$$

where k_b and k_a are nonnegative tuning coefficients, and z_I is the vector corresponding to the inertial z-axis.

These equations can be reduced based on Assumption 1. If the bias is constant, only one estimation is needed. Then the bias can be removed without the additional dynamics in Equation (3.2). A common practice is to estimate the bias by averaging a set of samples while the vehicle is at rest. [19].

3.4 Tuning

The estimated rotation is constructed primarily from the integration of angular rates, augmented by the acceleration vector, which can prevent small biases from being integrated into rotational drift. The parameters k_b and k_a can be thought as the cut-off frequencies in the linear filter with units of $\frac{\text{rad}}{sec}$, as discussed in [16]. As such, a higher k_a allows the acceleration measurement to contribute more high frequency information to the estimate of rotation. However, this is often undesirable since the high frequency noise in the accelerometer can disturb the angle measurement. If k_a is too low, the filter will behave more like integrated angular velocity.

3.5 Performance

The performance of the filter is verified with benchtop tests. In these tests, the vehicle is tilted by hand and the estimated angles are compared to orientation. This type of testing verifies performance in the best case scenario, where noise is at a minimum and the angle can be easily verified. The effectiveness of this estimator is demonstrated in both simulations and with flight data. In these two cases, the sensor data and the estimated Euler angles are shown.

In Figure 3.1, sensor data for a set of three simulated rotations is plotted. The first occurs at 2 seconds around the vehicle x-axis. The vehicle is rotated 23°, held at that angle for two seconds and then rotated back to 0°. The same maneuver is then performed around the vehicle y and z-axes at 12 and 26 seconds. In Figure 3.3 a similar procedure is performed on the test vehicle. The angular displacement is 55° and the rotations are performed around the y, x and z-axes in that order. It can be seen that the noise levels in the simulation are greater but do not include the large disturbances that correspond to impacts of the real vehicle. Values of k_a up to 10 performed well in simulation but a conservative choice of $k_a = 1$ is chosen for implementation in the test vehicle.

Figures 3.2 and 3.4 show the filter output, $\hat{R}_B^I$, converted to Euler angles. These outputs correspond to the maneuvers described in the paragraph above. In the simulations, the difference between the measured and the actual displacement is within 5°. In the flight data, the difference is within 10°.

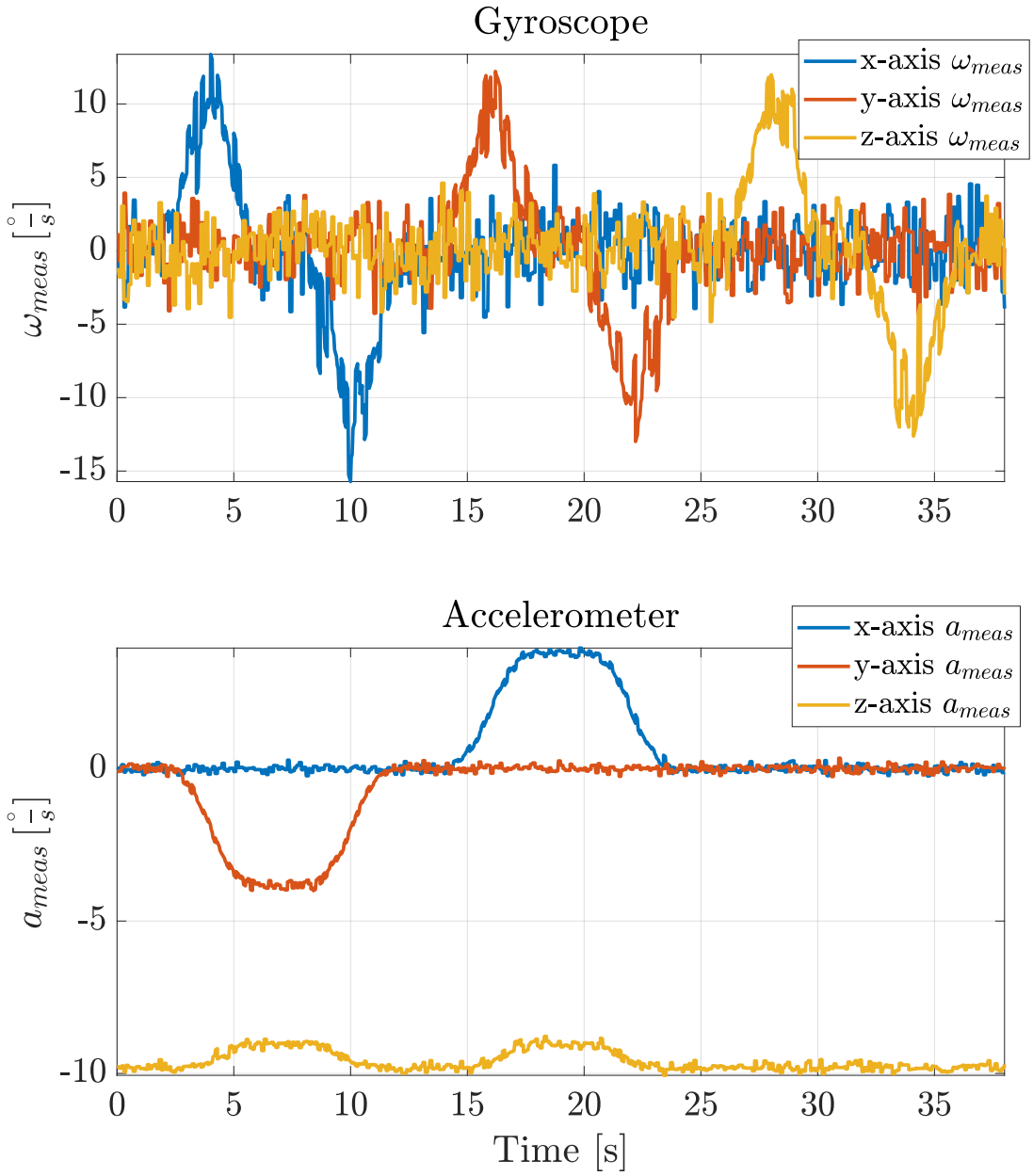

Figure 3.1: Sensor measurements during simulated estimator test.

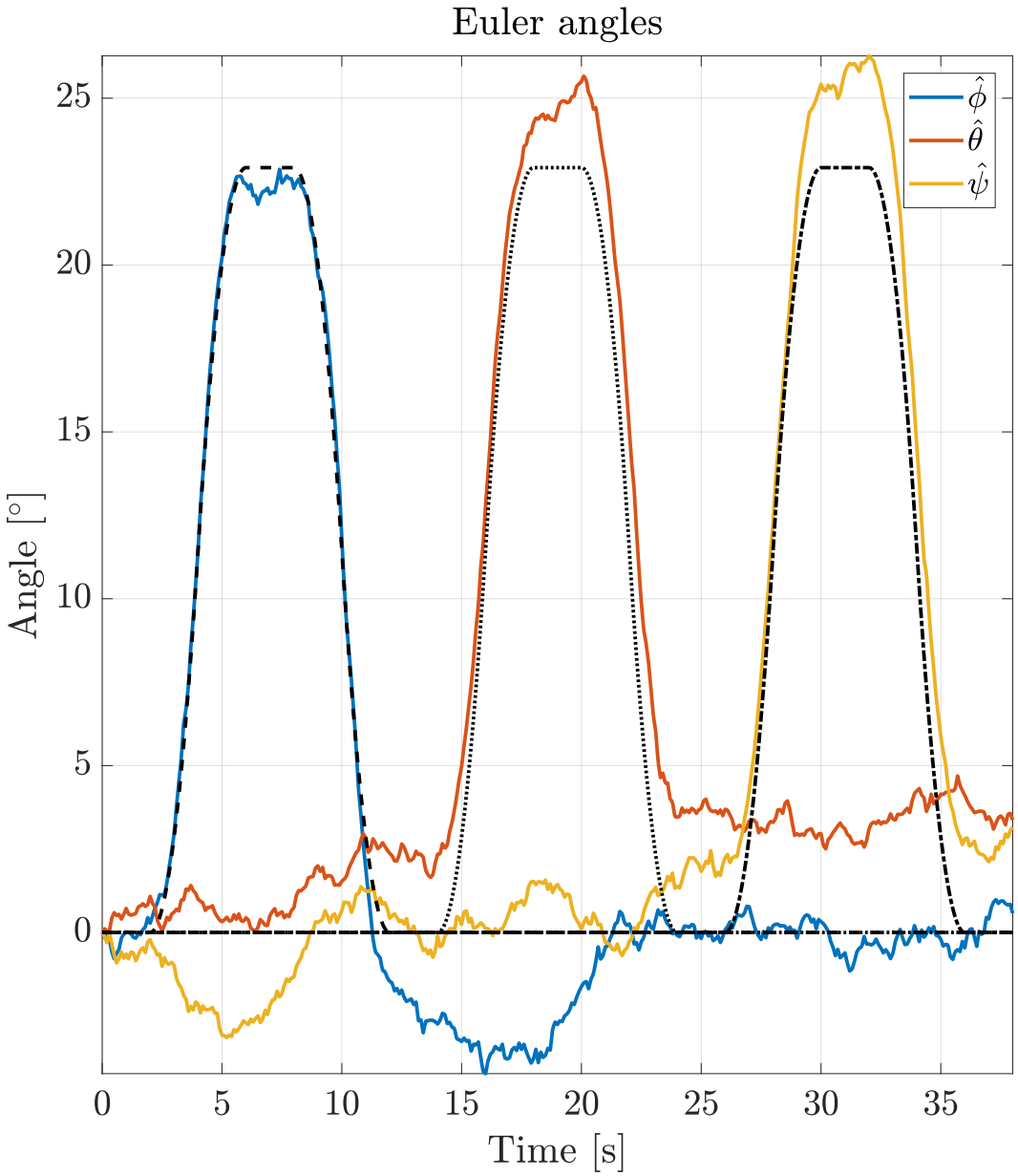

Figure 3.2: Orientation estimate during simulated estimator test, converted to Euler angles.

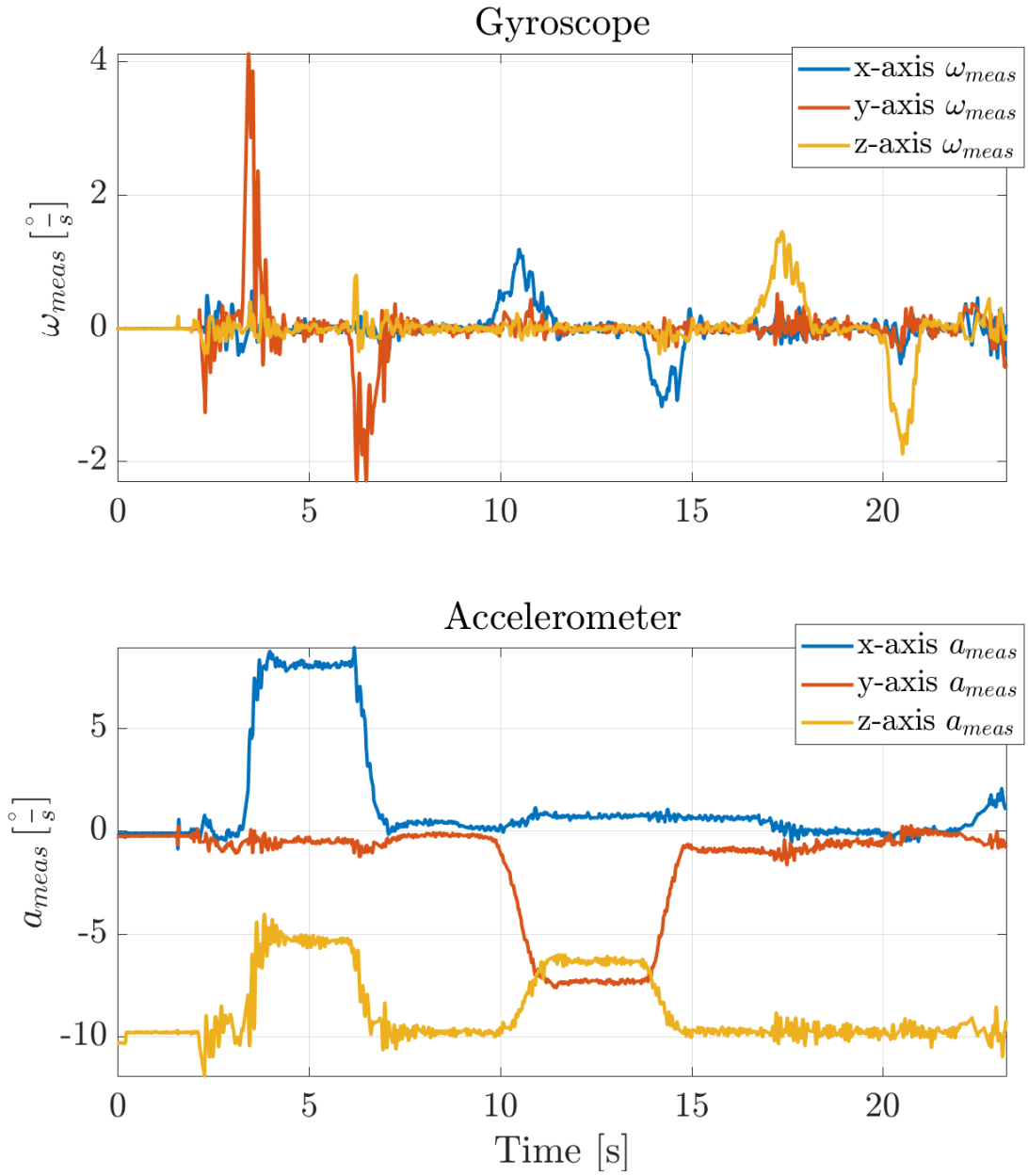

Figure 3.3: Sensor measurements during test vehicle estimator test.

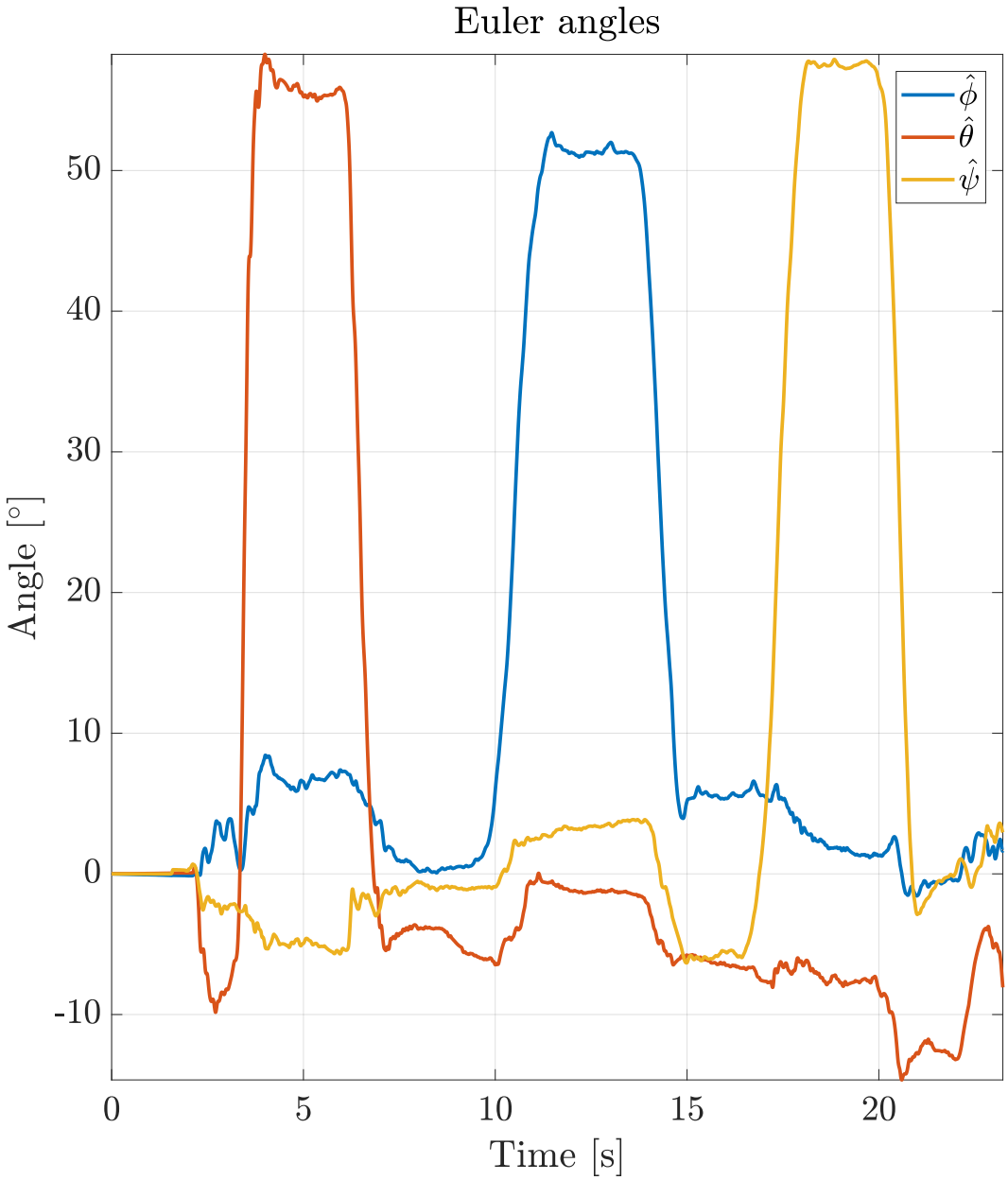

Figure 3.4: Test vehicle orientation estimate during estimator test, converted to Euler angles.

Chapter 4

Control

In order to perform system identification, we must have a controller that stabilizes the aircraft for human use. Once stabilized, data collection flights can be performed. Two controllers are introduced, which allow the system to be stabilized and can be extended when different, and more complex, control objectives are desired.

This chapter first introduces a cascaded PID controller, which uses one controller for every tracking objective. In our case, there will be two: one for angular rates, one for angles. If we were to use a position tracking controller, a third controller would be added and tuned. This control style is ubiquitous and often comes with commercial quadrotors. For this reason, PID controller is included for the sake of completeness.

There are several drawbacks to using this type of controller. First is the cascaded nature, which adds 3 tunable parameters for each control objective. In this case, there are six tuning parameters. The second problem is that this controller is designed based on a linearization of the dynamics of the quadrotor. As such, it is only accurate in a small region. If a disturbance causes a large angular displacement, the controller responds based on a poor approximation leading to oscillations or loss of control.

The second controller avoids these problems by mapping all the desired forces to the current state in one single equation set. As we will see, this controller is analogous to a PD controller that operates directly on SO(3), allowing it to provide the correct motor commands in all vehicle orientations [16].

4.1 Objective

Given the quadrotor dynamics:

$$\dot{\mathbf{x}} = g(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{4.1}$$

subject to the constraints

$$\omega < \omega_{max}$$
$$a < a_{max},$$

the control objective is to generate a set of moments to track a desired angular orientation, Θ_d and a desired angular velocity, ω_d. The human pilot should directly control the thrust of the vehicle since there is no sensor measurement of the altitude.

4.2 Controllers

4.2.1 PID

The PID controller presented is cascaded. As such, each level can be described separately. The first controller takes our desired attitude and compares it to the estimated attitude in the world frame. Based on the difference, it calculates a desired angular rate command and

attempts to track the angular rate with the second controller. The errors are given by

$$\Theta_{\mathrm{e}} = \Theta_{\mathrm{d}} - \hat{\Theta} = \begin{bmatrix} \phi_{\mathrm{d}} - \hat{\phi} \\ \theta_{\mathrm{d}} - \hat{\theta} \\ \psi_{\mathrm{d}} - \hat{\psi} \end{bmatrix} = \begin{bmatrix} \phi_{\mathrm{e}} \\ \theta_{\mathrm{e}} \\ \psi_{\mathrm{e}} \end{bmatrix}.$$

The angle controller calculates the desired angular rates with the following equations:

$$\begin{aligned} p_d &= k_{p1}\phi_{\mathrm{e}} + k_{i1}\int_0^t \phi_{\mathrm{e}}\mathrm{dt} + k_{d1}\frac{\mathrm{d}\phi_{\mathrm{e}}}{\mathrm{dt}} \\ q_d &= k_{p2}\theta_{\mathrm{e}} + k_{i2}\int_0^t \theta_{\mathrm{e}}\mathrm{dt} + k_{d2}\frac{\mathrm{d}\theta_{\mathrm{e}}}{\mathrm{dt}} \\ r_d &= k_{p3}\psi_{\mathrm{e}} + k_{i3}\int_0^t \psi_{\mathrm{e}}\mathrm{dt} + k_{d3}\frac{\mathrm{d}\psi_{\mathrm{e}}}{\mathrm{dt}}, \end{aligned}$$

where k_{px},k_{ix}, k_{dx} are the PID gains for the corresponding angles, and t is the current time. The angular rate errors can be represented as

$$\omega_{\mathrm{e}} = \omega_{\mathrm{d}} - \hat{\omega} = \begin{bmatrix} p_{\mathrm{d}} - p \\ q_{\mathrm{d}} - q \\ r_{\mathrm{d}} - r \end{bmatrix} = \begin{bmatrix} p_{\mathrm{e}} \\ q_{\mathrm{e}} \\ r_{\mathrm{e}} \end{bmatrix}.$$

The angular rate controller calculates the desired moments to be mapped to motor speed with the following equations:

$$\begin{aligned} u_{xd} &= k_{p4}p_{\mathrm{e}} + k_{i4}\int_0^t \phi_{\mathrm{e}}\mathrm{dt} + k_{d4}\frac{\mathrm{d}p_{\mathrm{e}}}{\mathrm{dt}} \\ u_{yd} &= k_{p5}q_{\mathrm{e}} + k_{i5}\int_0^t q_{\mathrm{e}}\mathrm{dt} + k_{d5}\frac{\mathrm{d}q_{\mathrm{e}}}{\mathrm{dt}} \\ u_{zd} &= k_{p6}r_{\mathrm{e}} + k_{i6}\int_0^t r_{\mathrm{e}}\mathrm{dt} + k_{d6}\frac{\mathrm{d}r_{\mathrm{e}}}{\mathrm{dt}}, \end{aligned}$$

where k_{px},k_{ix}, k_{dx} are the PID gains for the corresponding angular rates. In implementation, the derivative is often approximated with a backwards difference divided by the time step,

and the integral is approximated with a running sum. The running sum starts at takeoff and is multiplied by the time-step. These approximations are shown in the equations

$$\int_0^t x(\tau)\mathrm{dt} \approx \sum_{i=0}^{k} x[i]\mathrm{T},$$
$$\frac{\mathrm{d}x(t)}{\mathrm{dt}} \approx \frac{x[k]-x[k-1]}{\mathrm{T}},$$

where $x[i]$ is the value of the variable x at the i^{th} clock cycle tick, k is the current clock cycle time and T is the step time.

4.2.2 Geometric Control

The geometric quadrotor controller operates directly on the rotation matrix of the vehicle. Instead of relying on calculated Euler angles, it can compute the desired force and moments of the vehicle directly. However, since the operator is human, the target orientation will be given in Euler angles and used to construct the desired orientation, R_I^D. Given $[\phi_d, \theta_d, \psi_d]^\top$, we calculate

$$R_I^D = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - c_\phi s_\psi & s_\psi s_\phi + c_\psi c_\phi s_\theta \\ c_\theta s_\psi & c_\psi c_\phi + s_\psi s_\theta s_\phi & c_\phi s_\psi s_\theta - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}^\top . \tag{4.2}$$

The rotation in body frame, $\mathcal{B}$, is related to the desired frame, $\mathcal{D}$, as

$$R_B^D = R_I^D \hat{R}_B^I \tag{4.3}$$

where $\hat{R}_B^I$ is our estimated rotation in the world frame. The derivative of this orientation is given by

$$\dot{R}_B^D = \hat{R}_B^I \left(\omega - R_D^B \omega_\mathrm{d}\right), \tag{4.4}$$

which simplifies to

$$\dot{R}_B^D = \hat{R}_B^I \omega \tag{4.5}$$

since ω_{d} is zero.

Error Calculation

The attitude and angular rate tracking errors are given by the following equations:

$$e_R = \frac{1}{2}\left(R_B^D - R_D^B\right)_\vee$$
$$e_\omega = R_D^B \dot{R}_B^D = \omega_B - R_D^B \omega_D \Rightarrow e_\omega = \omega_B.$$

The desired moment output $u_m = [u_x, u_y, u_z]^T$ is then given by

$$u_m = -K_R e_R - K_\omega e_\omega, \tag{4.6}$$

where K_R and K_ω are positive definite diagonal gain matrices.

4.3 Tuning

The PID controller tuning methods are described in [20], so they will not be covered in this section. Instead, the focus will be on tuning the geometric control constants, K_R and K_ω. Since this controller is similar to a PD controller for small deflections from a horizontal orientation, these gains can be thought of as analogs to k_p and k_d as shown in [16]. Without the integral term, there are already fewer gains to tune and instead of cascading the gains, the gains each act directly on the measured angles and angular velocities.

The test vehicle is close to being symmetric, so we take the gain matrices K_R and K_ω to be scalars k_R and k_ω. If the tracking is poor in one direction due to this assumption, the gains can be expanded to $K_R = k_R \mathbb{I}_3$ and $K_\omega = k_\omega \mathbb{I}_3$ and the individual responses can be tuned separately.

The purpose of this controller is attitude tracking, with a secondary goal of tracking angular

velocity. By choosing k_R greater than k_ω, orientation tracking will take precedence. Simulated performance for well-chosen gains is shown in Figure 4.1.

Let these well-chosen gains be called k_R^* and k_ω^*. Then, for a poorly tuned controller with $k_R \gg k_R^*$, the response due to orientation error should be faster but more prone to overshoot. Since the overshoot is due to the estimated orientation and vehicle dynamics, the oscillations induced will produce medium frequency oscillations around the natural frequency for the vehicle. These oscillations can be seen in Figure 4.2 and can be improved by lowering k_R or increasing k_ω, which will dampen the oscillations by resisting the large angular velocities created.

Figure 4.3 shows a controller with $k_\omega \gg k_\omega^*$. In this case, the response to a change in target angle is sluggish. Another problem that may occur when the k_ω term is high is that noise can enter the system. Since k_ω directly multiplies ω, a large k_ω can amplify this noise. The result is that high frequency (frequencies greater than the vehicle dynamic frequencies) noise enters the system. Often this noise is too high of a frequency for the motors to track completely but can be heard when the motors attempt to track the change in desired speed. This high frequency noise can induce chattering in the vehicle.

These cases show how the system will behave when the proportion of k_R and k_ω is incorrect. If both are too high, the system will quickly go unstable. If both are too low, the system will struggle to respond, which may induce very low frequency oscillation around the target orientation, because it does not have the bandwidth to stop the overshoot.

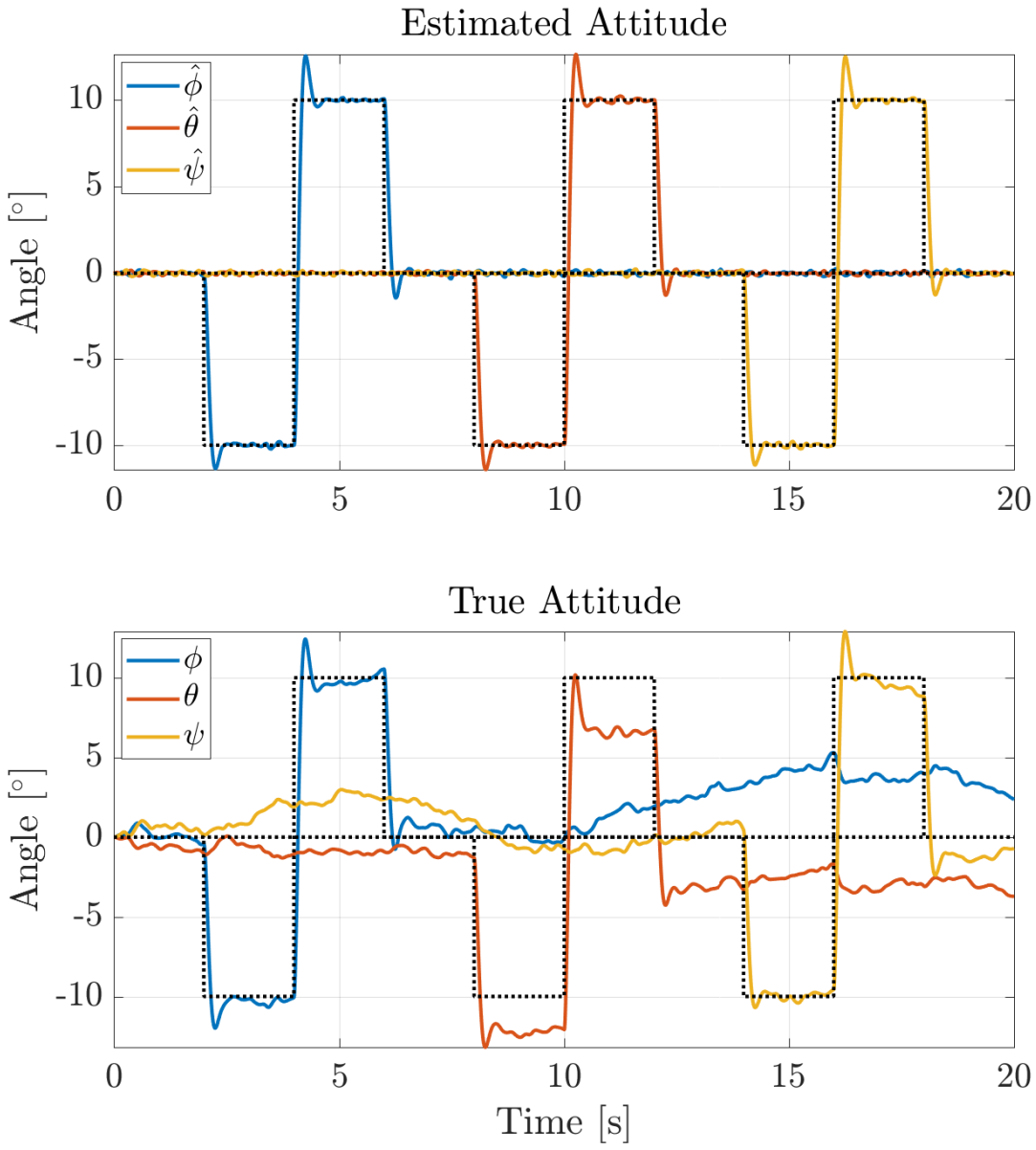

Figure 4.1: Estimated and true roll, pitch and yaw angle tracking of simulated vehicle with standard control tuning.

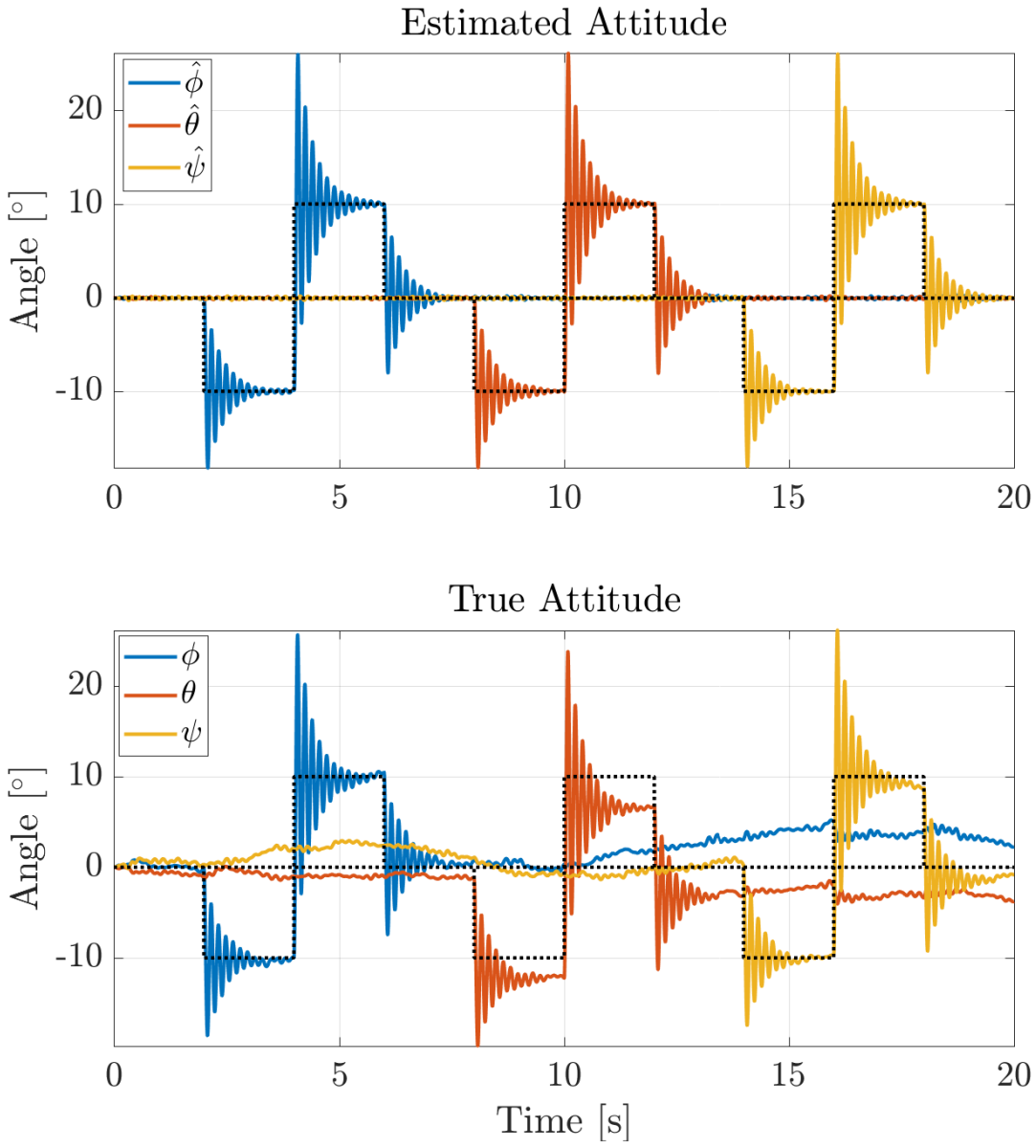

Figure 4.2: Estimated and true roll, pitch and yaw angle tracking of simulated vehicle when $k_R \gg k_R^*$.

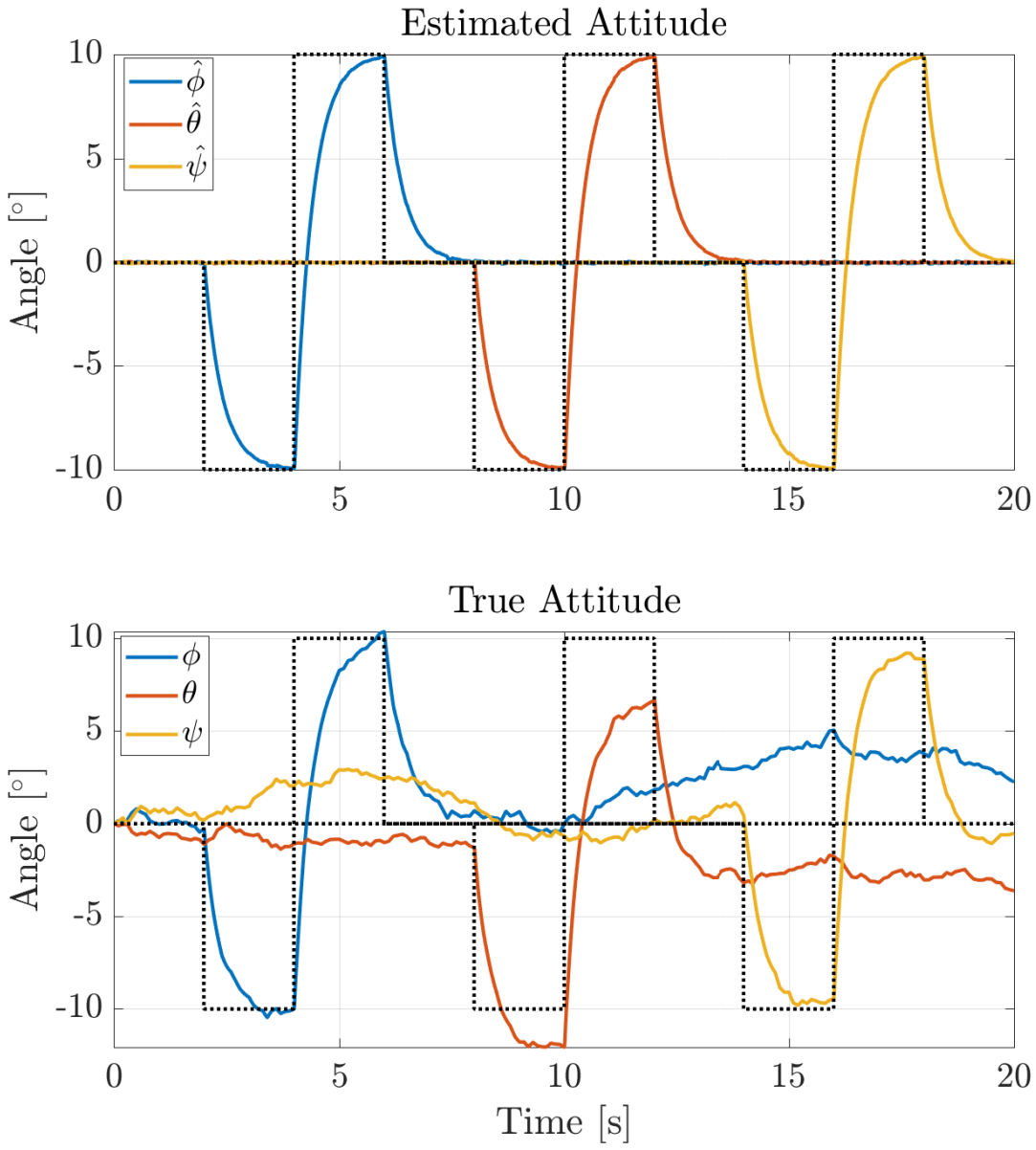

Figure 4.3: Estimated and true roll, pitch and yaw angle tracking of simulated vehicle when $k_\omega \gg k_\omega^*$.

4.4 Performance

Both the PID and geometric controllers are able to control the vehicle. The simulated performance can be seen in the plots above and the experimental performance is shown in Figure 4.4. The commands tracked in the figure are pitch commands of 4°, followed by a 28° command, and an 8° command.

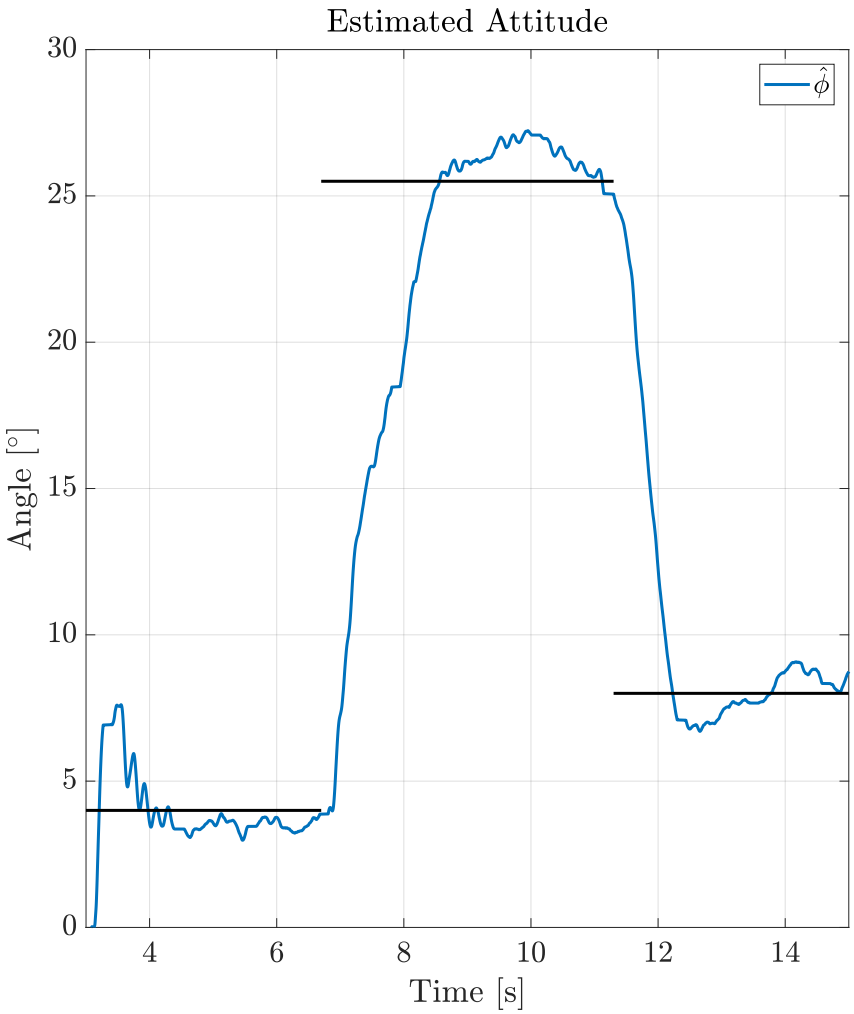

Figure 4.4: Estimated pitch angle tracking of real vehicle tracking 4° command

Chapter 5

System Identification

Now that the vehicle has an estimate of its state and is flying, the process of system identification can be started. This process will allow a simple 6 degree-of-freedom model to be estimated. With this model, more advanced controllers can be designed and simulations can be created to test performance. This chapter will outline how to estimate model parameters based on the assumption that the mass of the vehicle is known, and the minimum and maximum thrust of the motors are known.

5.1 Motor Thrust Parameters

There are several different ways to generate a thrust curve for a vehicle. The most detailed curves sample many different motor speed commands, and the corresponding forces are measured. These are called static tests, where the propeller and motor are fixed to a static mount with load cells to measure torque and motor moments. These stands are commercially available. Homemade test stands can be created for a single motor out of a scale. These test stands can provide reasonable accuracy but may not allow the range and precision of testing. If a stand is not available, the max thrust may be approximated by calculating the theoretical thrust based on the properties of the motors and propeller.

Recall that the thrust generated by each of the vehicle motors is given by

$$f_k = \begin{bmatrix} b_2 & b_1 & b_0 \end{bmatrix} \begin{bmatrix} \zeta_k^2 \\ \zeta_k \\ 1 \end{bmatrix}, \tag{5.1}$$

where b_0, b_1, $b_2 \geq 0$ and ζ_k is the k^{th} speed command. For k speed commands, the k force outputs are given by the equation

$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{bmatrix} = \begin{bmatrix} \zeta_1^2 & \zeta_1 & 1 \\ \zeta_2^2 & \zeta_2 & 1 \\ \vdots & & \\ \zeta_k^2 & \zeta_k & 1 \end{bmatrix} \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix}, \tag{5.2}$$

or $\mathbf{f} = \mathbf{X}\mathbf{b}$, where $\mathbf{X}$ is known as the Vandermonde matrix. To estimate the coefficients in $\mathbf{b}$, we can solve the least squares problem and find that

$$\mathbf{b} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{f}. \tag{5.3}$$

In the case of the test vehicle, three measurements are used to construct the thrust curve (with units of gram-force). The first data point is at a speed command of zero, where the thrust is zero. The second point is the hover speed command, which was tested by flying the vehicle in place. The output force is then equal to the mass of the vehicle. The last point is calculated by measuring the thrust of a single motor and scaling by the number of motors. Combined, these measured values yield:

$$\begin{bmatrix} 0 \\ 31 \\ 56 \end{bmatrix} = \begin{bmatrix} 0^2 & 0 & 1 \\ 0.5951^2 & 0.5951 & 1 \\ 0.95^2 & 0.95 & 1 \end{bmatrix} \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix}. \tag{5.4}$$

Solving this equation using the least mean squared solution in Equation (5.3) produces

estimated polynomial coefficients of $\mathbf{b} = [0.0022, 0.3835, 0]^T$. The results of this estimation are compared against the estimate performed by the company that created the test vehicle in Figure 5.1. It can be seen that the thrust curves do not vary significantly, especially in the takeoff and hover range.

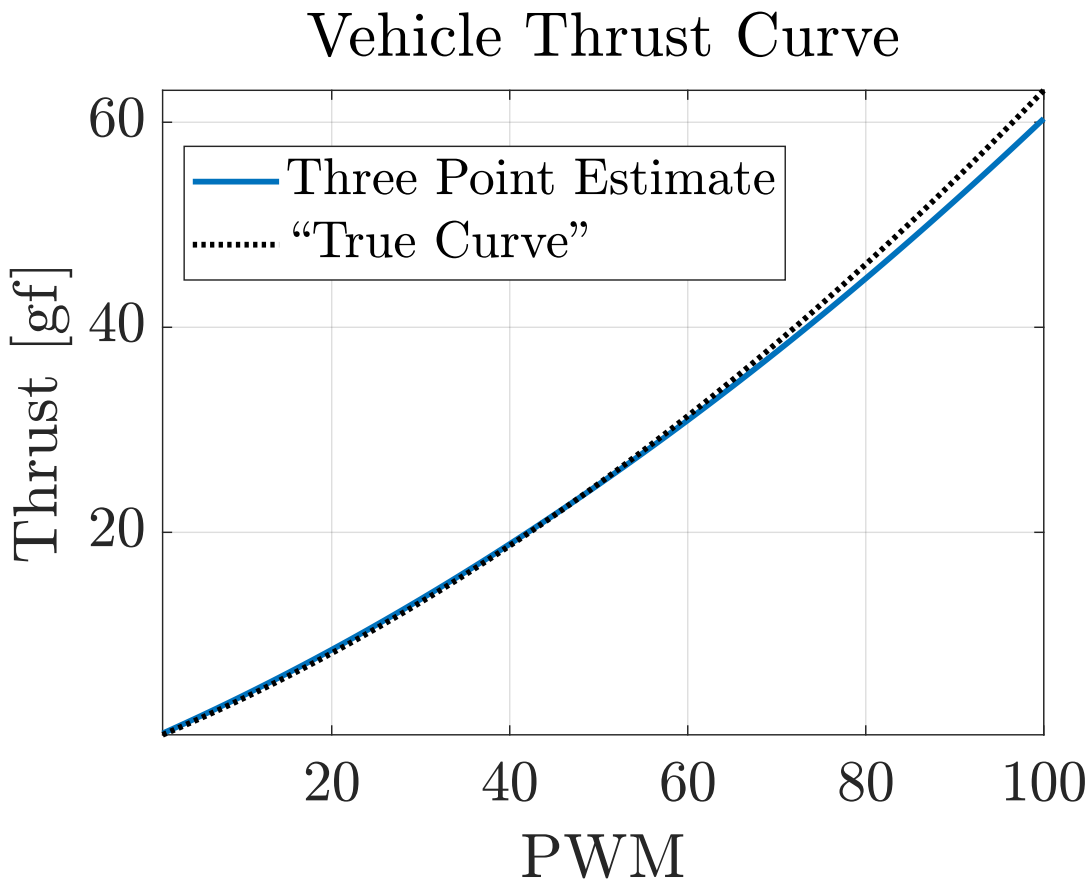

Figure 5.1: The dotted curve represents the measurements performed by [21].

5.2 Inertia Identification

Given the calculated torque applied to the vehicle and the measured angular velocities, this section discusses how to construct an estimate of the vehicle's inertia matrix. First, a new formulation for the rotational dynamics is introduced based on [22]. Then two methods of solving the system of equations are presented. The first is a naive least squares approach based on the work of [7]. The second approach is developed to solve a constrained version of the problem, taking into account the fact that the inertia matrix cannot have negative elements.

5.2.1 Attitude Dynamics

Consider the equation of motion for a rigid body with torque inputs:

$$\dot{\omega} = \mathcal{J}^{-1}\left[\tau - \omega \times (\mathcal{J}\omega)\right]. \tag{5.5}$$

To find a least squares solution, this equation needs to be converted to the form $\mathbf{A}x = \mathbf{b}$. To this end, the following matrices are defined:

$$\Omega = \begin{bmatrix} \omega_1 & 0 & 0 & \omega_2 & \omega_3 & 0 \\ 0 & \omega_2 & 0 & \omega_1 & 0 & \omega_3 \\ 0 & 0 & \omega_3 & 0 & \omega_1 & \omega_2 \end{bmatrix} \tag{5.6}$$

$$\mathrm{J} = \begin{bmatrix} \mathcal{J}_{xx} & \mathcal{J}_{yy} & \mathcal{J}_{zz} & \mathcal{J}_{xy} & \mathcal{J}_{xz} & \mathcal{J}_{yz} \end{bmatrix}. \tag{5.7}$$

The rigid body dynamics can then be re-written as

$$\tau = \left[\dot{\Omega} + \omega \times \Omega\right] \mathrm{J} \tag{5.8}$$

$$\int \tau \,\mathrm{dt} = \left[\Omega + \int \omega \times \Omega \,\mathrm{dt}\right] \mathrm{J}. \tag{5.9}$$

Combining these two equations we can express the dynamic equation as $\mathrm{Y} = H\mathrm{J}$, with

$$\mathrm{Y} = \begin{bmatrix} \tau \\ \int \tau \, \mathrm{dt} \end{bmatrix},$$
$$\mathrm{H} = \begin{bmatrix} \dot{\Omega} + \omega \times \Omega \\ \Omega + \int \omega \times \Omega \, \mathrm{dt} \end{bmatrix}.$$

5.2.2 Least Squares

With the linear equation $\mathrm{Y} = H\mathrm{J}$, the estimate of the inertia matrix can be calculated at each sample time as

$$\hat{\mathrm{J}} = (H^T H)^{-1} H^T \mathrm{Y}, \tag{5.10}$$

which minimizes the square of the residual of our estimate according to:

$$\hat{\mathrm{J}} = \arg\min_{\mathrm{J}} \|H\mathrm{J} - \mathrm{Y}\|^2. \tag{5.11}$$

For k time steps, the equation can be solved by creating H and Y matrices from each sample:

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_k \end{bmatrix} \qquad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_k \end{bmatrix}. \tag{5.12}$$

This method is useful for quick tests, but it is sensitive to noise. In cases with large noise levels, the matrix inversion $(H^T H)^{-1}$ is often poorly conditioned. This can result in solutions, that are orders of magnitude too large and may result in negative inertia estimates, which have no physical meaning.

5.2.3 Non-negative Least Squares

The previous method is constrained to produce symmetric results by the construction of the J vector. However, the methods do not preclude the possibility of negative solutions. The goal of the non-negative method is to find

$$\hat{\mathrm{J}} = \arg\min_{\mathrm{J}} \|H\mathrm{J} - \mathrm{Y}\|^2, \tag{5.13}$$

subject to

$$\mathrm{J} \geq 0. \tag{5.14}$$

A detailed algorithm for generating this solution is given in [8], and implementations can be found in many scientific programming languages. It can also be solved as a nonlinear optimization problem in the full form or as a quadratic programming problem by writing

$$\hat{\mathrm{J}} = \arg\min_{\mathrm{J}\geq 0} \frac{1}{2}\mathrm{J}^T H^T H\mathrm{J} - \mathrm{Y}^T H\mathrm{J}. \tag{5.15}$$

This solution can be implemented in a way that avoids the numerical problems of the naive least squares estimation algorithm and is used to estimate the inertia of the vehicle.

5.3 Multi-sine Test Input

In order to ensure conditions that allow the regression equations to have enough information to converge to a solution, the vehicle must undergo some amount of forced motion. The multi-sine technique provides a framework for generating this forced motion. The technique used to design test inputs for our vehicle is outlined in [23].

The technique generates a set of orthogonal multi-sine input signals. These signals are added to the vehicle inputs in order to perturb the dynamics. The ideal input covers the entire bandwidth of the actuator so that all possible outputs are modeled without overexciting, and possibly damaging, the actuator. Therefore, the input to one actuator should produce a response orthogonal to that of the other outputs in both the time and frequency domain.

Each of these test inputs is a sum of sinusoids with unique amplitudes, frequencies and phase shifts. For the j^{th} actuator, the test input is given by

$$\mathbf{v}_j = \sum_{k \in \{1,2,\ldots,M\}} A_k \sin\left(\frac{2\pi k t}{T} + \phi_k\right), \tag{5.16}$$

where M is the number of distributed sine waves, t is the time vector, T is the duration of the excitation, A_k is the k^{th} amplitude, and ϕ_k is the k^{th} phase shift.

The sinusoidal frequencies $\omega_k = 2\pi k/T$ are chosen based on the bandwidth of the actuator and based on the distribution of frequencies to keep frequency separation with the other actuators. For example, if there are four actuators and each has an actuator bandwidth of 1 Hz, then a possible frequency distribution with four test frequencies for each actuator would be to equally distribute the test signals, as shown below: With this distribution, each actuator is excited throughout the frequency range.

Actuator	Frequencies			
1	0.1	0.34	0.58	0.82
2	0.16	0.4	0.64	0.88
3	0.22	0.46	0.7	0.94
4	0.28	0.52	0.76	1.0

Table 5.1: Example distribution of frequencies across four actuators.

The phase shifts for the sine waves, ϕ_k, are chosen so that the relative peak factor of the multi-sine input v_j is minimized. This peak factor, given by the equation

$$\mathrm{RPF}(\mathbf{v}_j) = \frac{\max(\mathbf{v}_j) - \min(\mathbf{v}_j)}{2\sqrt{2}\mathrm{rms}(\mathbf{v}_j)}, \tag{5.17}$$

is minimized to prevent constructive interference in the sine waves from creating an input that drives the system too far away from the desired operating conditions.

The amplitude of each sine wave is chosen to keep a uniform power distribution based on the formula

$$A_k = \frac{A}{\sqrt{n}}, \tag{5.18}$$

where A is the desired power level. A signal-to-noise ratio of 10 is often chosen to insure adequate information.

An example test input, generated using SIDPAC from [23], for the four quadrotor motors is shown in Figure 5.2 with $A = 5$, a frequency range of 0.1 Hz to 2 Hz. The relative peak factors on all four signals are below 1.2 and contain 15 different frequencies distributed in the frequency range. These multi-sines can also be generated in MATLAB using the `idinput` command or be hand generated by adding sine waves manually.

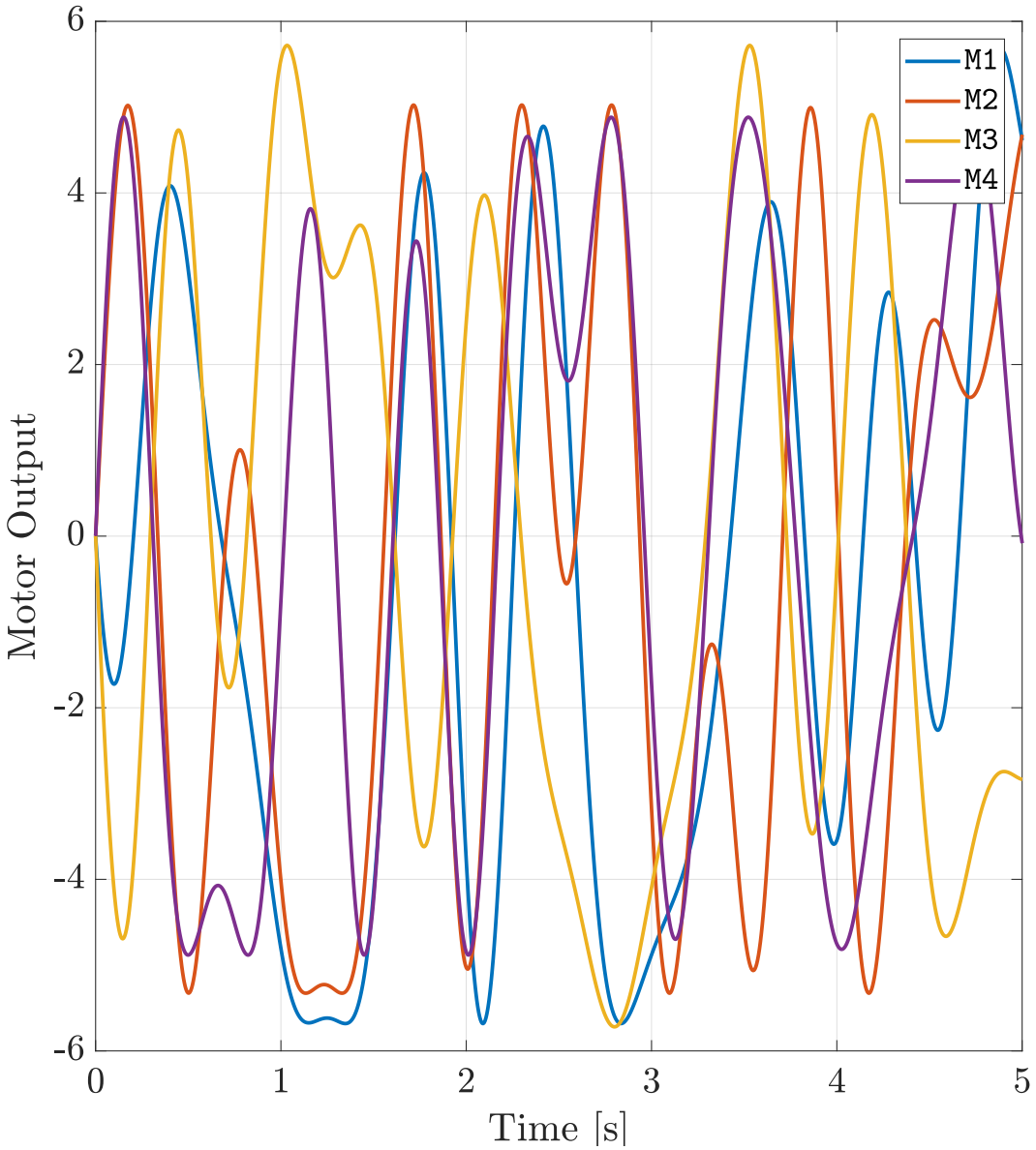

Figure 5.2: Test inputs for four motors.

Chapter 6

Results

The previous chapter covered the generation of test inputs and two methods of inertia identification. This chapter presents the results of the application of these methods along with a comparison of effectiveness.

To measure effectiveness we will look at the convergence of the estimate to the true value, the speed of convergence and numerical stability while varying input type and the method of approximating the inertia.

For clarity, only the principle axes of the inertia matrix will be shown, since the off-diagonal elements are small. This subset of the inertia matrix will be denoted $\mathbf{J} = [\mathcal{J}_{xx}, \mathcal{J}_{yy}, \mathcal{J}_{zz}]$. The metric for accuracy will be the 2-norm of the difference between the actual inertia and the estimated inertia

$$E = \|\mathbf{J} - \hat{\mathbf{J}}\|_2. \tag{6.1}$$

Both simulation results and implementation results are demonstrated in this chapter. The simulation is performed with a reference inertia matrix of $\mathcal{J} = \mathbb{I}$. The vehicle inertia is expected to be similar to the inertia measured in [12]

$$[\mathcal{J}_{xx}, \mathcal{J}_{yy}, \mathcal{J}_{zz}] \approx 10^{-5}[2.5, 2.5, 3.2]\text{kg m}^2.$$

6.1 Inertia Estimation with No-Input Excitation

In this case, the vehicle tries to estimate the inertia of the vehicle, excited only by the input required to reject angular disturbances. The following figures show the measured attitude, the estimated inertia using the non-negative least squares estimation technique and the residual comparison between the least squares method and the non-negative least squares method. Since the estimation technique occasionally produces extremely large estimates, the output is averaged. Note that in Figure 6.3 the non-negative estimate converges to a residual around 0.2, while the signal-to-noise ratio causes too many incorrect estimates for the standard least squares algorithm to converge.

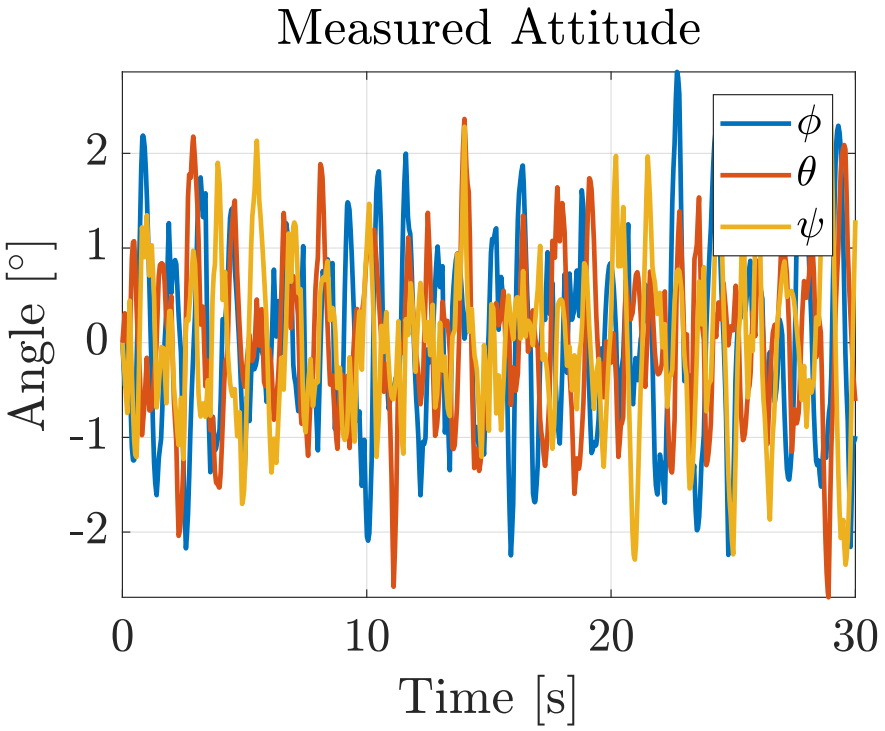

Figure 6.1: Angle measurements during inertia estimation with zero reference.

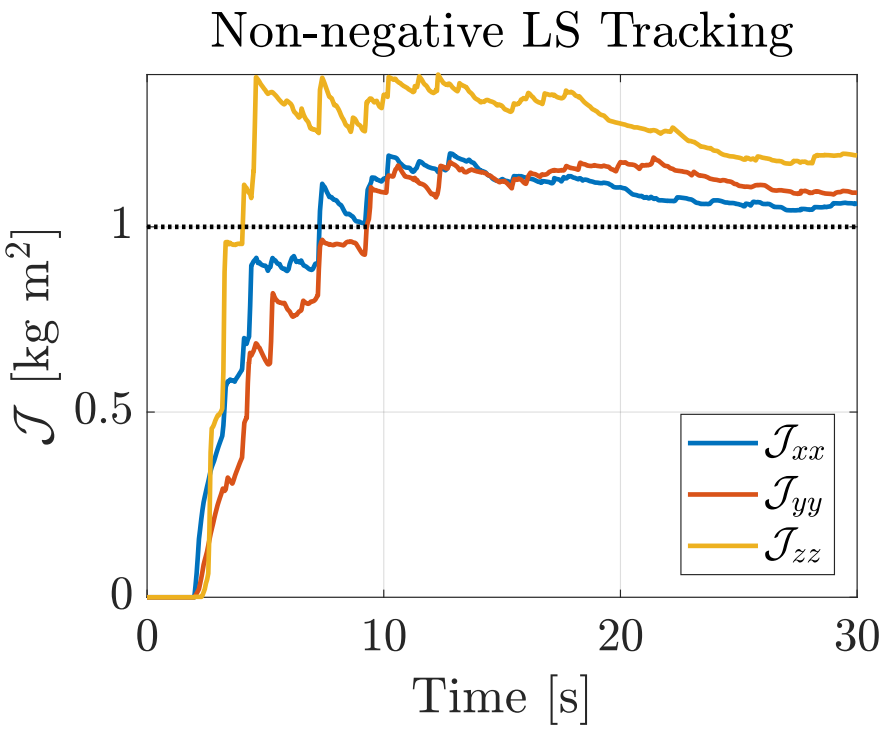

Figure 6.2: Average of inertia estimates after 30s.

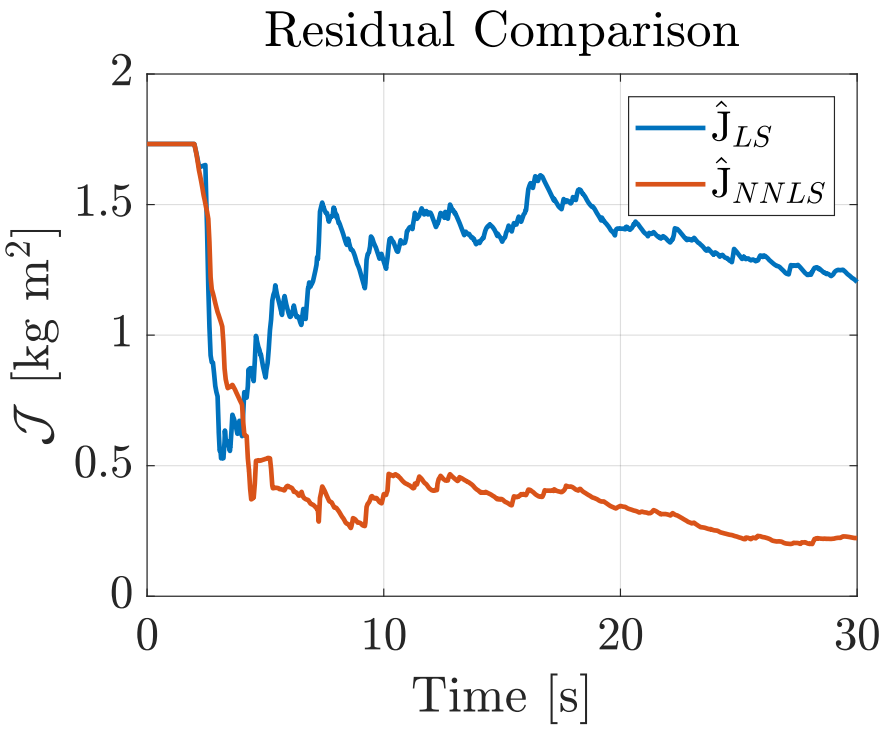

Figure 6.3: Residual comparison of the LS and Non-negative LS estimations.

6.2 Inertia Estimation with Single Sine Excitation

In this case the vehicle tries to estimate the inertia of the vehicle, excited by only a 10°, 1 $\frac{\text{rad}}{\text{sec}}$ reference. The following figures show the measured attitude, the estimated inertia using the non-negative least squares estimation technique and the residual comparison between the least squares method and the non-negative least squares method. In Figure 6.6 the non-negative estimate converges to a residual around 0.1 and the traditional least squares approach does get closer to the true value than the no input case.

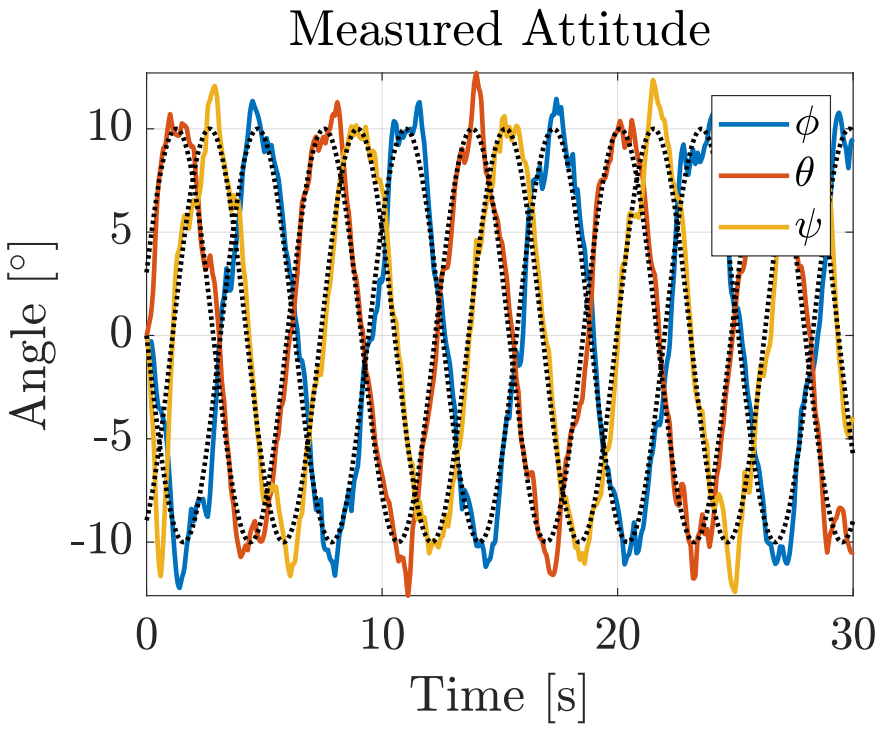

Figure 6.4: Angle measurements during inertia estimation with a 10°, 1 $\frac{\text{rad}}{\text{sec}}$ reference.

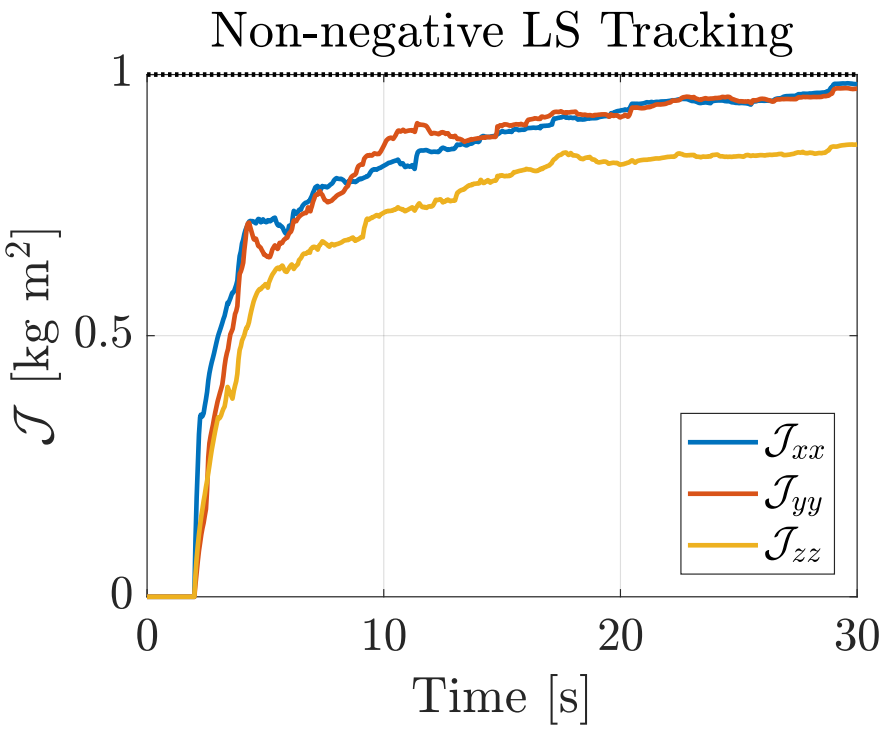

Figure 6.5: Average of inertia estimates after 30s with a 10°, 1 $\frac{\text{rad}}{\text{sec}}$ reference.

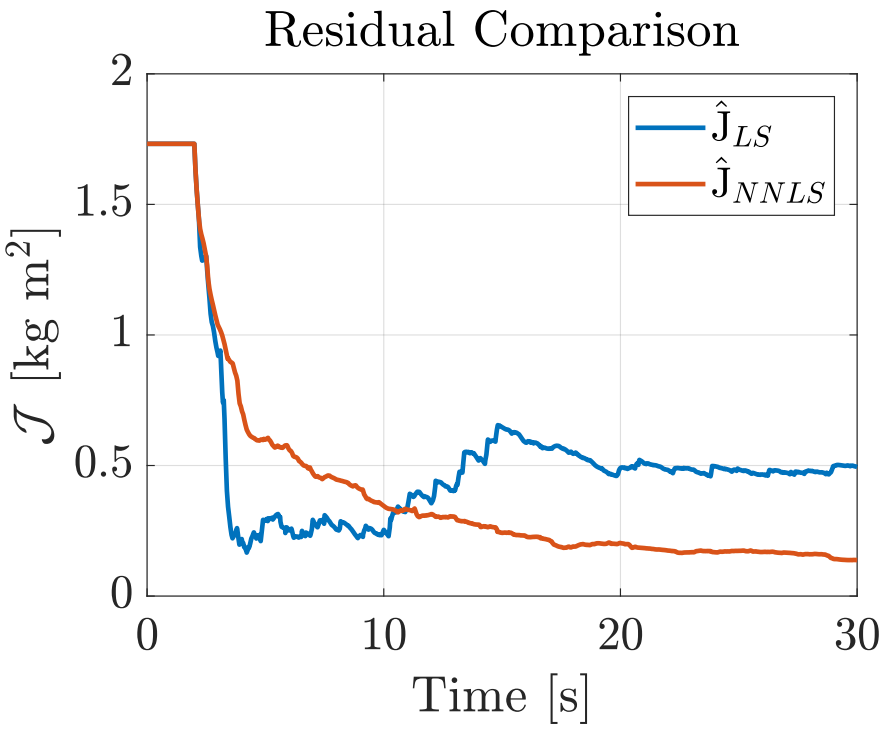

Figure 6.6: LS and Non-negative LS residuals with a 10°, 1 $\frac{\text{rad}}{\text{sec}}$ reference.

In this case the vehicle tries to estimate the inertia of the vehicle, excited by only a 25°, 1 $\frac{\text{rad}}{\text{sec}}$ reference. The following figures show the measured attitude, the estimated inertia using the non-negative least squares estimation technique and the residual comparison between the least squares method and the non-negative least squares method. Note that in Figure 6.9, both of the estimates converge. The least squares estimate appears to converge faster, possibly because of the variance in solutions causing the average to shift faster.

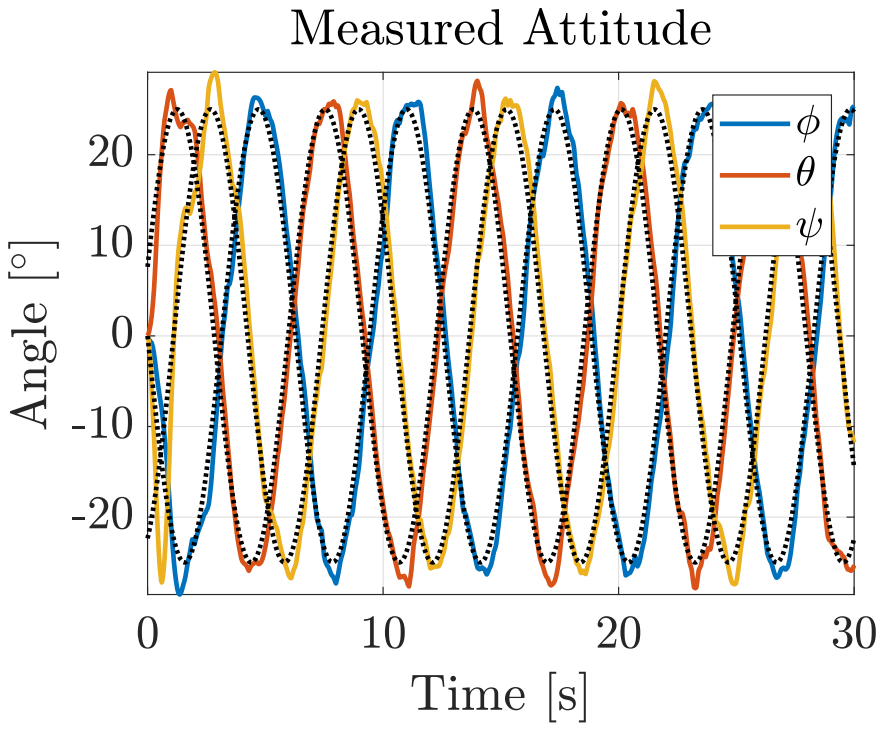

Figure 6.7: Angle measurements during inertia estimation with a 25°, 1 $\frac{\text{rad}}{\text{sec}}$ reference.

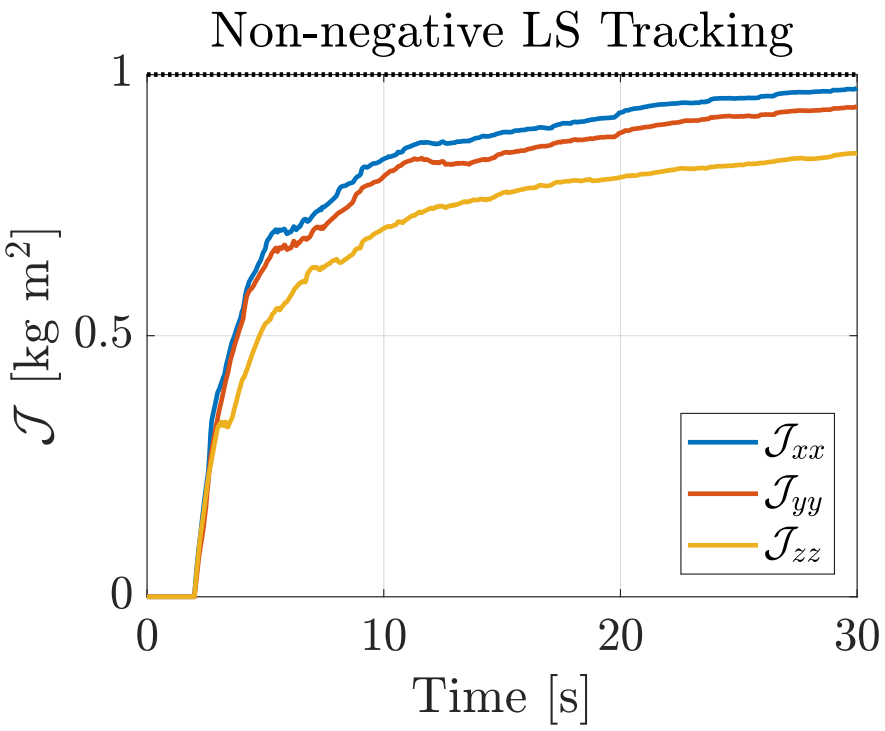

Figure 6.8: Average of inertia estimates after 30s with a 25°, 1 $\frac{\text{rad}}{\text{sec}}$ reference.

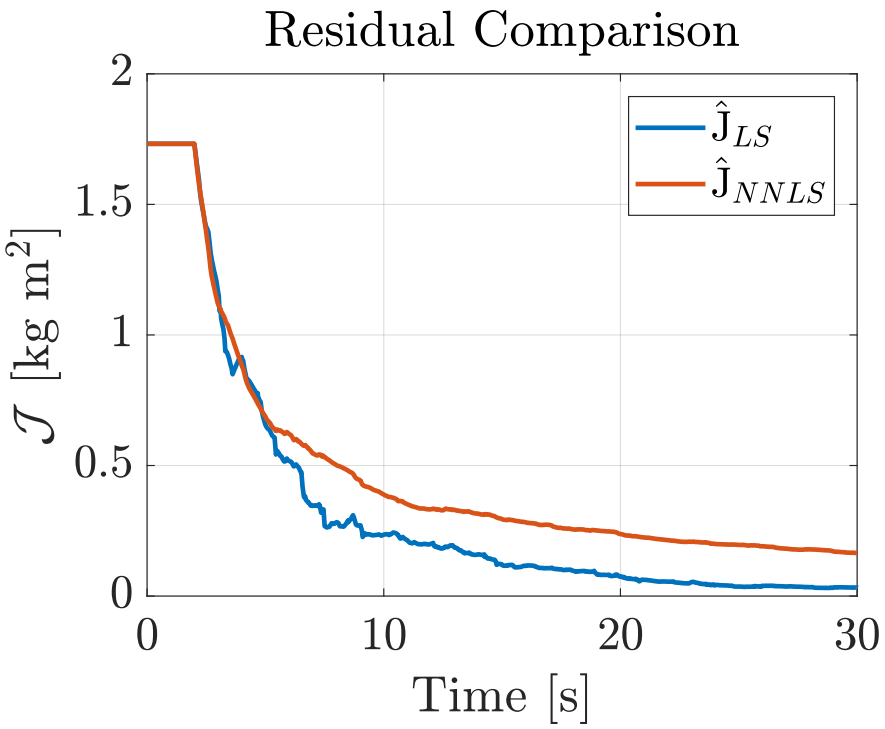

Figure 6.9: LS and Non-negative LS residuals with a 25°, 1 $\frac{\text{rad}}{\text{sec}}$ reference.

6.3 Inertia Estimation Multisine

In this case the vehicle tries to estimate the inertia of the vehicle, excited by a 10° multisine input similar to the one shown in the previous chapter. The following figures show the measured attitude, the estimated inertia using the non-negative least squares estimation technique and the residual comparison between the least squares method and the non-negative least squares method. Figure 6.12 shows that even for a smaller angular input, the extra frequency information allows the residuals to converge as in the 25° single sine case.

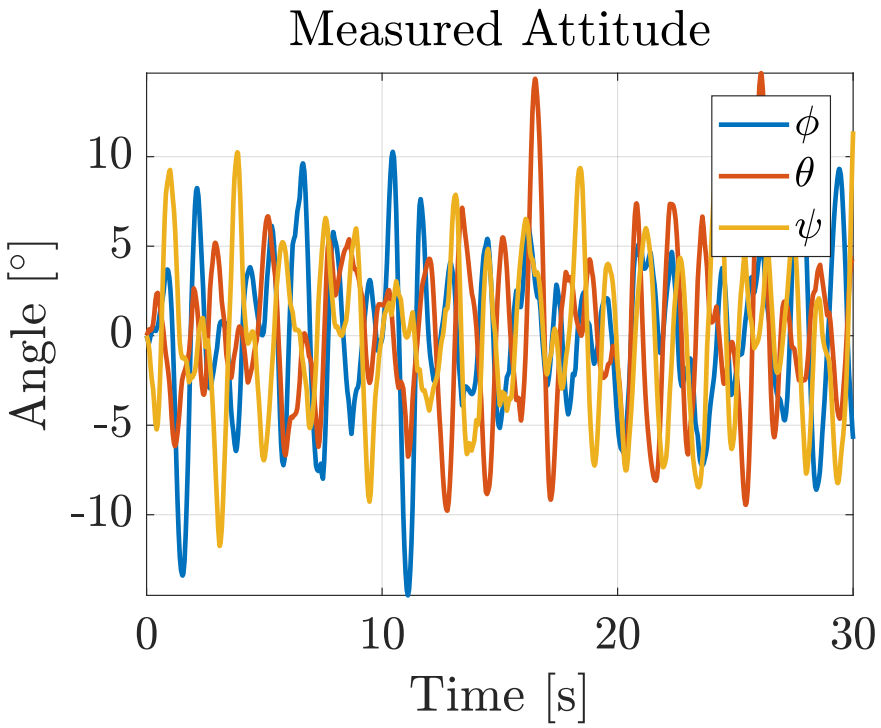

Figure 6.10: Angle measurements during inertia estimation with a 10° multisine reference.

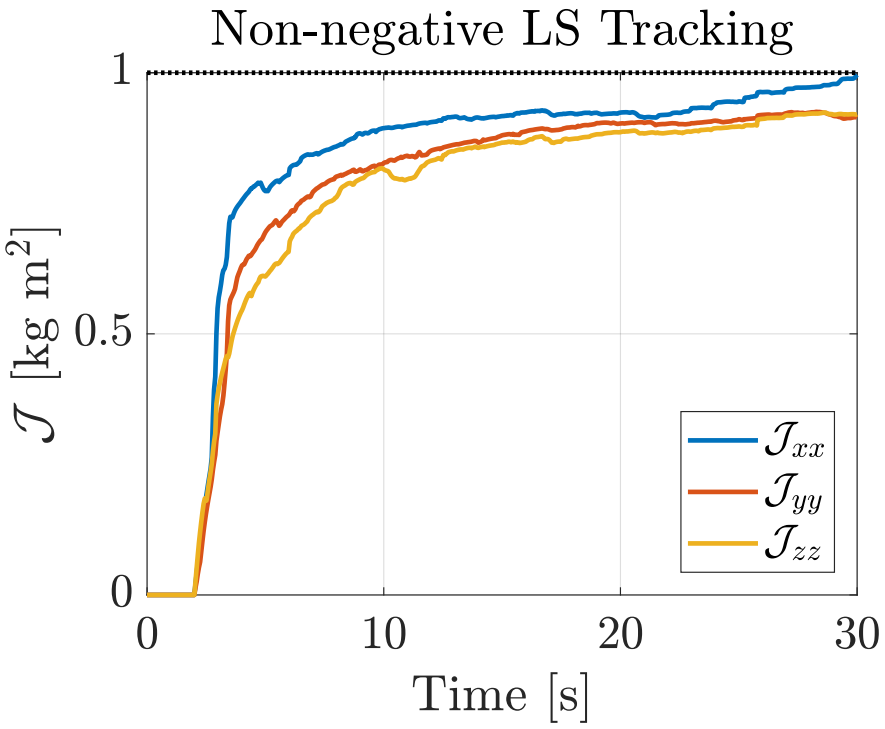

Figure 6.11: Average of inertia estimates after 30s with a 10° multisine reference.

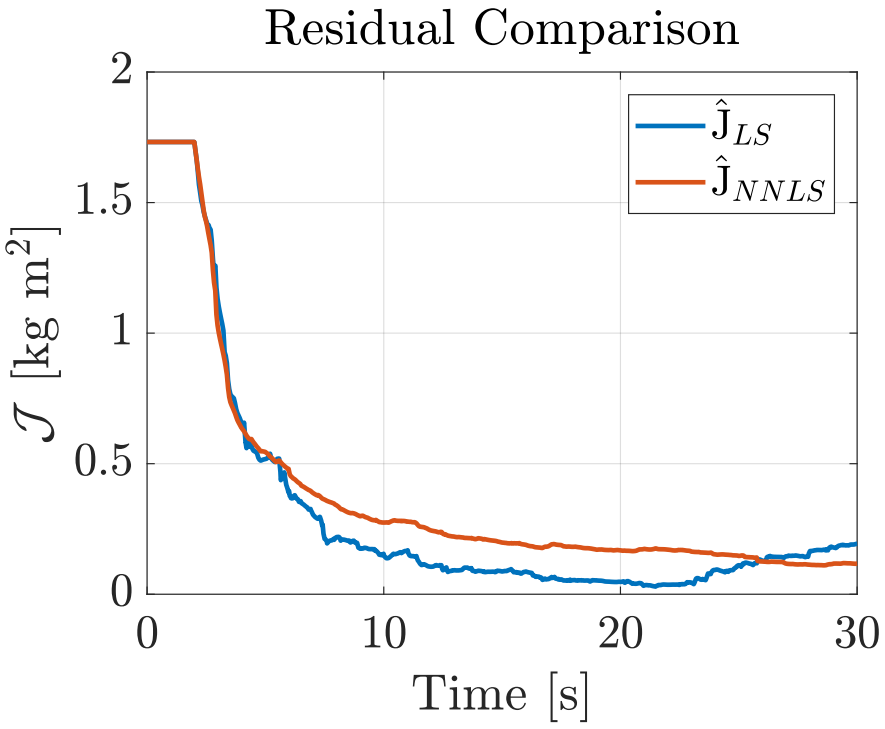

Figure 6.12: LS and Non-negative LS residuals with a 10° multisine reference.

In this case the vehicle tries to estimate the inertia of the vehicle, excited by a 20° multisine input similar to the one shown in the previous chapter. The following figures show the measured attitude, the estimated inertia using the non-negative least squares estimation technique and the residual comparison between the least squares method and the non-negative least squares method. Figure 6.15 shows the best convergence and both the nonnegative and the unconstrained least squared estimates converge.

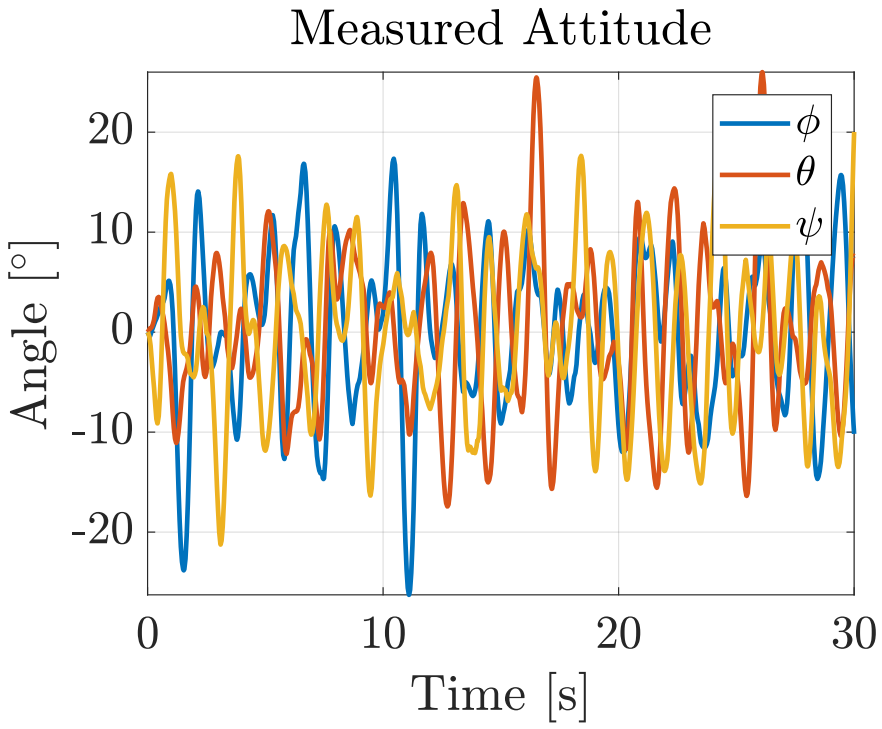

Figure 6.13: Angle measurements during inertia estimation with a 20° multisine reference.

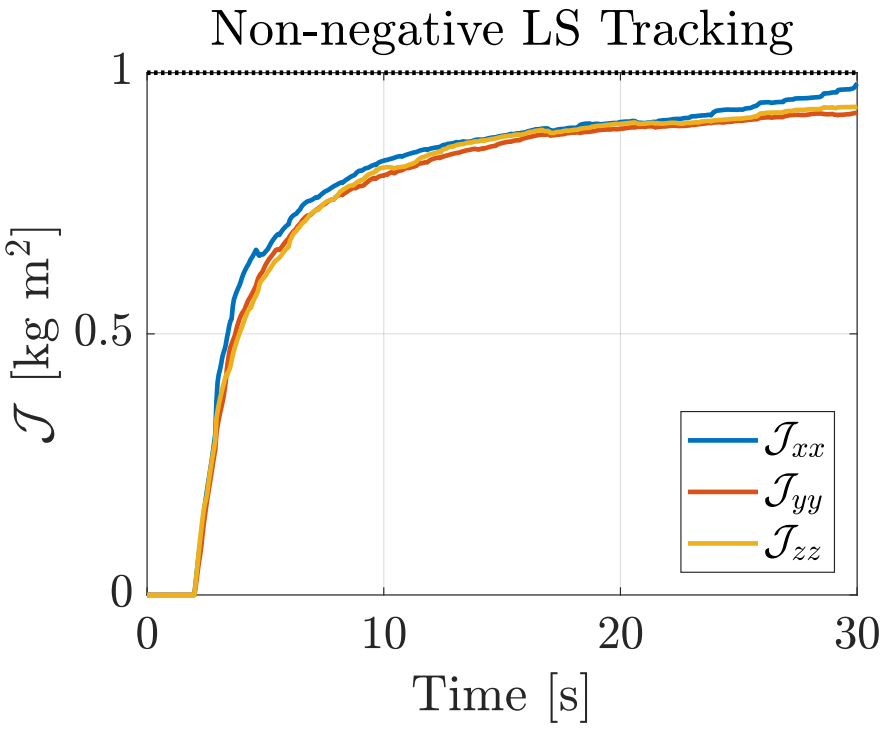

Figure 6.14: Average of inertia estimates after 30s with a 20° multisine reference.

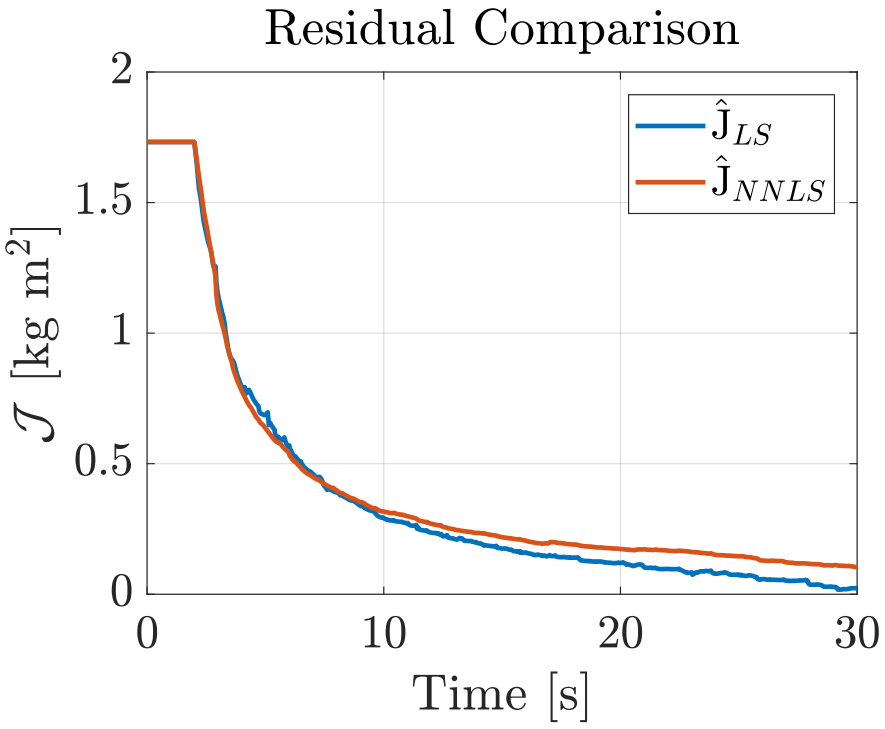

Figure 6.15: LS and Non-negative LS residuals with a 20° multisine reference.

6.4 Implementation Results

In implementation, only the non-negative least squares algorithm is used since it proved to be the most stable in the presence of noise. Figure 6.16 shows the instantaneous inertia estimate for a 10-second flight with the primary inertias plotted. The noise contribution from both the commanded torques and measured angular velocities makes the inertia estimates difficult to process and many spikes appear in the data. A zoomed-in view is shown in Figure 6.17. Note that believable estimates of the inertia vary but do not include massive peaks or zeros for the main diagonal elements of the inertia matrix.

To condition the data, first, data spikes larger than two orders of magnitude greater than the expected inertia range are removed. Secondly, zeros resulting from poor excitation are removed. The remaining information is combined through averaging, resulting in primary inertias of $\mathbf{J} = [2.26,\ 2.23,\ 2.87] \cdot 10^{-5}$. The third component is not plotted for clarity in the graphs.

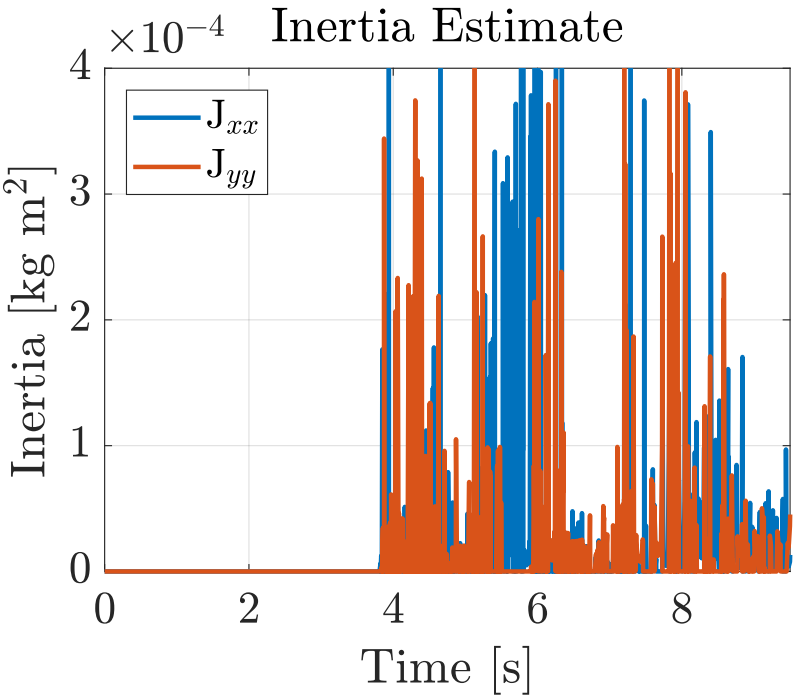

Figure 6.16: Time history of inertia estimates for a quadrotor flight.

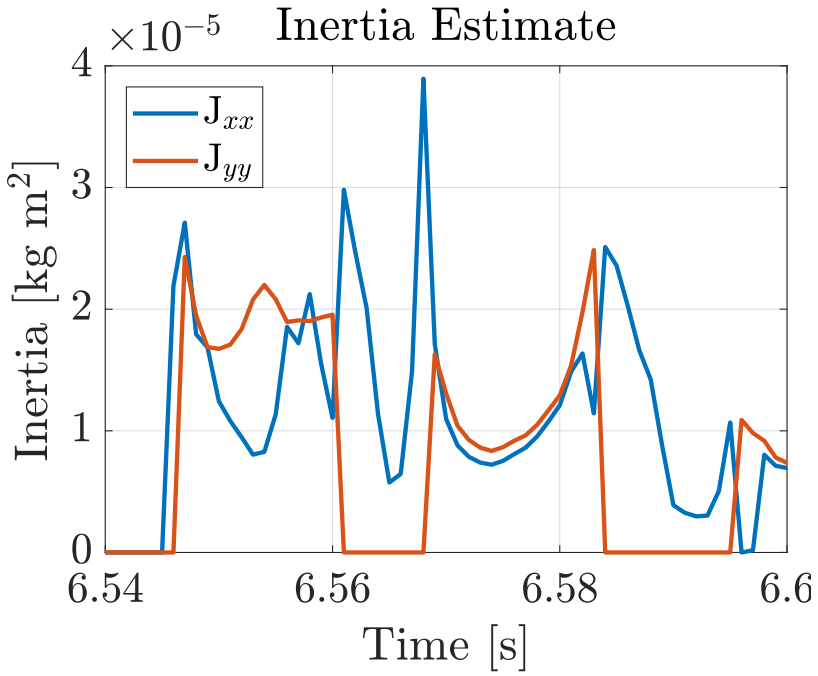

Figure 6.17: Detailed view of Inertia estimate with spikes removed.

Chapter 7

Conclusion

This thesis presents a new method of estimating inertia for distributed electric vehicles, which extends least squares methods to include the nonlinear constraint of non-negativity. In chapter 2, the equations of motion for the quadrotor are discussed and a test vehicle is selected. Chapter 3 presents a state estimator design for SO(3). In chapter 4, two controller designs are presented to ensure vehicle stability during the identification process. In chapter 5, a system identification approach for inertia is formulated and test inputs are generated to aid convergence. In chapter 6, the results of the non-negative least squares method are compared with the unconstrained method both in simulation and in flight tests. In both cases, the non-negative constraint led to a better estimate of the inertia than traditional least squares.

Future Work

The amount of noise in the implementation made application of the least squares algorithm difficult and required more hand tuning as compared to simulations. There are three main avenues towards improvement. The first is to use a larger vehicle for demonstration. The vehicle used is small; the inertia is close to zero and so the thrust produced by the motors can be quickly buried in the noise. The second avenue of improvement is the addition of a rudimentary positioning system. Though it takes away from the simplicity of the

estimation, it is hard to keep the vehicle stable during system identification, especially given the large perturbations when flying by hand. The third direction of improvement is in the controller and estimator design. The estimation of $\dot{\omega}$ is a common challenge. Many different solutions have been proposed, but the feedback of the high frequency angular rate noise into the controller made even the control signal hard to use for estimation. Combining the modifications to the test vehicle, a less naive estimate of $\dot{\omega}$ with a control tuning less sensitive to noise should yield far better implementation results.

Bibliography

[1] Directorate-General for Mobility and Transport. White paper on transport. Technical report, 2011.

[2] Ehang 184 autonomous aerial vehicle. `http://www.ehang.com/ehang184`. Accessed: 2017-4-6.

[3] Jeff Holden and Nikhil Goel. Fast-Forwarding to a future of On-Demand urban air transporation. Technical report, 27 October 2016.

[4] Geoffrey Bower. Vahana configuration trade study part II vahana, 1 February 2017. Accessed: 2017-4-6.

[5] Joseph Chambers. The role of dynamically scaled Free-Flight models in support of NASA's aerospace programs.

[6] Kathy Barnstorff. Ten-Engine electric plane completes successful flight test. 30 April 2015.

[7] Donghoon Kim, Sungwook Yang, and Sangchul Lee. Rigid body inertia estimation using extended kalman and savitzky-golay filters. *Math. Probl. Eng.*, 2016, 1 June 2016.

[8] Charles Lawson and Richard Hanson. *Solving least squares problems*. SIAM, 1995.

[9] Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall PTR, 1999.

[10] Matthew Rich. *Model development, system identification, and control of a quadrotor helicopter*. PhD thesis.

[11] Mohammad Alsharif and Matthew Hölzel. System identification of a quadcopter's rotational dynamics using android flight data. In *2016 IEEE Conference on Control Applications (CCA)*, pages 786–791, 2016.

[12] Benoit Landry. *Planning and Control for Quadrotor Flight through Cluttered Environments*. PhD thesis, Massachusetts Institute of Technology, June 2015.

[13] Jared Grauer. Real-Time parameter estimation using output error.

[14] Prasad Hemakumara and Salah Sukkarieh. Non-parametric UAV system identification with dependent gaussian processes. In *2011 IEEE International Conference on Robotics and Automation*, pages 4435–4441, May 2011.

[15] Moses Bangura, Robert Mahony, and Others. Nonlinear dynamic modeling for high performance control of a quadrotor. In *Australasian conference on robotics and automation*, pages 1–10, 2012.

[16] Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robot. Autom. Mag.*, 19(3):20–32, 2012.

[17] Jan Vervoorst. A modular simulation environment for the improved dynamic simulation of multirotor unmanned aerial vehicles, 2016.

[18] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Trans. Automat. Contr.*, 53(5):1203–1218, June 2008.

[19] Mark Loony. A simple calibration for MEMS gyros. *EDN Europe*, 2010.

[20] Karl Åström and Tore Hägglund. PID controllers: Theory, design, and tuning. *The International Society of Measurement and Control*, 1995.

[21] Bitcraze. Thrust investigations. `https://wiki.bitcraze.io/misc:investigations:thrust`. Accessed: 2017-4-16.

[22] Sungwook Yang, Sangchul Lee, Jung-Hyung Lee, and Hwa-Suk Oh. New Real-Time estimation method for inertia properties of STSAT-3 using gyro data. *Trans. Jpn. Soc. Aeronaut. Space Sci.*, 58(4):247–249, 2015.

[23] Eugene Morelli. Flight test maneuvers for efficient aerodynamic modeling. *J. Aircr.*, 49 (6):1857–1867, 2012.